

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## GENERÁTOR HODINOVÉHO SIGNÁLU

CLOCK GENERATOR

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Ondřej Dornák

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Zdeněk Bradáč, Ph.D.

BRNO 2020

# Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Ondřej Dorňák

**ID:** 195481

**Ročník:** 3

**Akademický rok:** 2019/20

**NÁZEV TÉMATU:**

## Generátor hodinového signálu

### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte experimentální generátor hodinového/obdélníkového signálu.

1. Proveďte literární a internetovou rešerši.
2. Navrhněte koncept generátoru na bázi TCXO nebo obdobném řešení.
3. Navrhněte a realizujte elektronický generátor s vyššími požadavky na frekvenční stabilitu v rozsahu frekvencí 1-20Hz. Výstup generátoru řešte jako proudově posílený. Pro generátor navrhněte schéma zapojení a zapojení DPS.
4. Výsledný systém zhodnoťte.

### DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6

Dle pokynů vedoucího práce.

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 8.6.2020

**Vedoucí práce:** doc. Ing. Zdeněk Bradáč, Ph.D.

**doc. Ing. Václav Jirsík, CSc.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Bakalářská práce se zabývá návrhem a realizací generátoru hodinového signálu na bázi teplotně kompenzovaného krystalu. Teoretická část se zabývá možnými zdroji hodinového signálu. Dále je pak stanoven koncept generátoru. V praktické části se nachází samotný návrh obvodu spolu s výpočty obvodových součástek.

## KLÍČOVÁ SLOVA

generátor, hodinový signál, teplotně kompenzovaný krystal, programovatelný logický obvod

## ABSTRACT

This thesis focuses on design and implementation of a clock generator based on temperature compensated crystal oscillator. The theoretical part deals with possible sources of clock signals. Next, the concept of the generator is determined. The practical part contains circuit design and component calculation.

## KEYWORDS

generator, clock source, temperature compensated crystal oscillator, programmable logic device

DORŇÁK, Ondřej. *Generátor hodinového signálu*. Brno, 2020, 63 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Generátor hodinového signálu“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Zdeňku Bradáčovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

# Obsah

<b>Úvod</b>	<b>11</b>
<b>1 Základní parametry generátorů</b>	<b>12</b>
<b>2 Zdroje signálu</b>	<b>14</b>
2.1 RC oscilátor s Wienovým článkem . . . . .	14
2.2 Integrované obvody . . . . .	15
2.2.1 LM555 . . . . .	15
2.2.2 ICL8038 . . . . .	16
2.3 Krystalové rezonátory . . . . .	16
2.3.1 TCXO . . . . .	18
2.3.2 OCXO . . . . .	19
<b>3 Generátor na bázi TCXO</b>	<b>20</b>
3.1 Logické obvody . . . . .	20
3.2 VHDL . . . . .	22
3.2.1 Základní struktura jazyka VHDL . . . . .	22
3.2.2 Paralelní příkazy . . . . .	23
3.2.3 Sekvenční příkazy . . . . .	24
3.3 Digitálně analogový převodník . . . . .	25
3.4 Návrh konceptu generátoru . . . . .	25
<b>4 Návrh generátoru</b>	<b>28</b>
4.1 Napájení a stabilizace . . . . .	28
4.2 TCXO . . . . .	29
4.3 CPLD . . . . .	30
4.4 Tlačítka . . . . .	31
4.5 Displej . . . . .	32
4.6 DA převodník . . . . .	34
4.7 Výstupní buffer . . . . .	35
4.8 Návrh DPS . . . . .	36
4.9 Krabice . . . . .	37
4.10 Programové řešení . . . . .	37
4.11 Osazení a oživení . . . . .	40
4.12 Měření parametrů . . . . .	41
<b>Závěr</b>	<b>43</b>

<b>Literatura</b>	<b>44</b>
<b>Seznam příloh</b>	<b>46</b>
<b>A Schéma zapojení</b>	<b>47</b>
<b>B DPS a osazovací schéma</b>	<b>49</b>
<b>C Rozpiska materiálu</b>	<b>50</b>
<b>D Části modelu krabičky</b>	<b>51</b>
<b>E Foto, osazená DPS a oživení</b>	<b>52</b>
<b>F Zdrojové soubory VHDL</b>	<b>53</b>
<b>G Obsah CD</b>	<b>63</b>



# Seznam obrázků

2.1	Oscilátor s wienovým článkem . . . . .	14
2.2	LM555 jako multivibrátor (převzato z [2] ) . . . . .	15
2.3	Zapojení ICL8038 (převzato z [3] ) . . . . .	16
2.4	Náhradní schéma krystalu a jeho použití . . . . .	17
2.5	Vnitřní struktura TCXO (převzato z [6] ) . . . . .	18
2.6	Kompenzace, graf (převzato z [7] ) . . . . .	18
2.7	OCXO (převzato z [8] ) . . . . .	19
3.1	Vnitřní struktura CPLD(převzato z [10]) . . . . .	21
3.2	Vnitřní struktura obvodu FPGA(převzato z [11]) . . . . .	21
3.3	Blokové schéma binární děličky . . . . .	26
3.4	Blokové schéma obvodu trojúhelníkového signálu . . . . .	26
3.5	Blokové schéma konceptu generátoru . . . . .	27
4.1	Schéma zapojení USB konektoru a stabilizátoru . . . . .	29
4.2	Schéma zapojení TCXO . . . . .	30
4.3	Zapojení CPLD . . . . .	31
4.4	Tlačítko s obvodem pro omezení zákmitů . . . . .	31
4.5	Zapojení displeje . . . . .	33
4.6	Zapojení DA převodníku . . . . .	35
4.7	Zapojení výstupního bufferu . . . . .	36
4.8	Sestava krabičky . . . . .	37
4.9	Blokové schéma, funkční popis programu . . . . .	38
4.10	Úprava napájecího napětí zesilovačů . . . . .	40

# Seznam tabulek

1.1	Základní parametry generátoru . . . . .	13
4.1	Parametry stabilizátoru . . . . .	28
4.2	TCXO, tabulka parametrů . . . . .	29
4.3	XC9572X1, tabulka parametrů . . . . .	30
4.4	SA39-12SRWA, tabulka parametrů . . . . .	32
4.5	BC807, tabulka parametrů . . . . .	33
4.6	MCP6021, tabulka parametrů . . . . .	36
4.7	Měření amplitudy signálů, tabulka naměřených hodnot . . . . .	41
4.8	Měření hran obdelníkového signálu, tabulka naměřených hodnot . . .	41
4.9	Měření periody signálů . . . . .	42

# Seznam výpisů

3.1	Struktura VHDL . . . . .	22
3.2	VHDL, strukturální popis . . . . .	23
3.3	Nepodmíněné přiřazení ve VHDL . . . . .	24
3.4	Čtyřbitový multiplexor ve VHDL . . . . .	24
3.5	VHDL, klopný obvod typu D . . . . .	25
F.1	reseni.vhd . . . . .	53
F.2	main_divider.vhd . . . . .	56
F.3	divider.vhd . . . . .	57
F.4	debouncer.vhd . . . . .	58
F.5	counter.vhd . . . . .	59
F.6	core.vhd . . . . .	60
F.7	seven_seg_decode.vhd . . . . .	62

# Úvod

Cílem této práce je navrhnout malý generátor hodinového signálu s vyššími nároky na frekvenční stabilitu. Práce je rozvržena do čtyř kapitol. V první kapitole jsou uvedeny parametry, důležité při výběru či návrhu generátoru. Druhá kapitola, na základě literární rešerše, nastiňuje možné zdroje hodinového signálu. Ve třetí kapitole se nachází teoretický rozbor pro návrh generátoru na bázi teplotně kompenzovaného krystalu (TCXO). Následuje poslední kapitola obsahující detailní návrh celého obvodu, včetně návrhu desky plošných spojů (DPS), spolu s výpočtem hodnot obvodových součástek a popisem programu pro programovatelný logický obvod. Dále se zde nachází postup při ožívání obvodu, včetně odstranění chyb v návrhu a následné měření parametrů zařízení. Celou práci zakončuje závěr s hodnocením dosažených výsledků.

Navrhovaný generátor bude sloužit jako součást laboratorního automatizovaného měřicího systému. Pro tento účel lze použít komerční funkční generátor, kterým daná laboratoř disponuje. Na něm lze nastavit požadované parametry v širokém rozsahu včetně spousty dalších nastavení nepodstatných pro měřicí systém. Tento systém běžně měří po poměrně dlouhou dobu a tedy znemožňuje využití funkčního generátoru pro jiná laboratorní měření. Jelikož se jedná o poměrně drahé zařízení, které disponuje mnoha funkcemi nepotřebných pro měřicí systém, je přistoupeno k návrhu malého generátoru pro daný systém.

# 1 Základní parametry generátorů

Generátor či funkční generátor je elektronické zařízení, jehož výstupem je elektrický signál definovaného tvaru. Tím může být například sinus, obdelník, trojúhelník, pila nebo jakýkoliv uživatelsky definovaný tvar. Nejčastěji se používají v laboratořích pro měření vlastností elektronických obvodů, nebo jako reference při kalibraci zařízení.

Generátory lze rozdělit do dvou základních skupin na základě principu jejich fungování, tedy na analogové a digitální. Analogové generátory mívají nejčastěji sinusový signál. Jsou využívány především pro jejich nízké zkreslení signálu a možnosti generování vysokých kmitočtů. Jejich nevýhoda je nižší stabilita. Naproti tomu digitální generátory bývají stabilnější a umožňují generovat i signály o velmi nízké frekvenci ( v řádech  $\mu Hz$ ) [1].

Parametrů, podle kterých se vybírá vhodný generátor je celá řada. Je potřeba přihlížet k potřebám dané aplikace. Například pro měření ve vysílací technice se přihlíží k sinusovému generátoru, možnost směšování signálů či modulace. Naopak v digitální technice se sáhne spíše po generátoru pravoúhlých signálů, stejně tak při použití generátoru při kalibracích, kde se očekává také vysoká přesnost a stabilita.

Seznam obecných základních parametrů vystihuje tabulka 1.1. Mezi další důležité parametry lze zařadit například možnost nastavení stejnosměrné složky signálu nebo možnost měnit střidu obdelníkových signálů.

Z všeobecného hlediska jsou důležitými parametry také rozměry celého zařízení, rozsah provozních teplot či přítomnost periférií umožňující propojení s měřícím systémem.

Tab. 1.1: Základní parametry generátoru

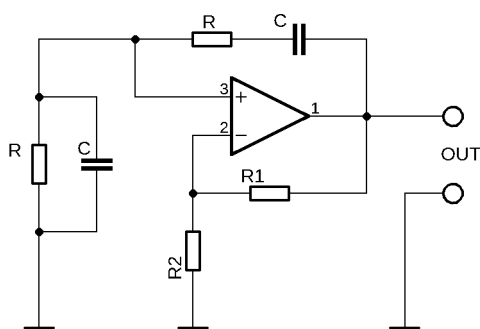
Název	Jednotka	Definice
Amplituda	$[V]$	Hodnota výstupního napětí, nebo rozsah hodnot, kterých může výstup nabývat
Tolerance Amplitudy	$[\%]$	Maximální možná odchylka výstupní od nastavené amplitudy
Jmenovitý kmitočet	$[Hz]$	Hodnota, nebo rozsah hodnot kmitočtu výstupního signálu
Kmitočtová stabilita	$[ppm]$	Odchylka výstupní frekvence od nastavené. Dělí se na krátkodobou ( $<1s$ ) a dlouhodobou ( $\gg 1s$ ).
Teplotní stabilita	$[ppm/^{\circ}C]$	Odchylka výstupní frekvence od nastavené při oteplení o $1^{\circ}C$
Zatěžovací impedance	$[\Omega]$	Minimální hodnota zátěže, při které je generátor schopen pracovat

## 2 Zdroje signálu

Na základě literární rešerše a průzkumu trhu jsou dále uvedeny některé z možností konstrukce generátoru. Uveden je oscilátor se selektivním RC článkem. Dále možnost použití integrovaných obvodů. Podkapitola o krystalových oscilátorech pojednává o principu činnosti krystalů a způsobu jejich teplotní stabilizace.

### 2.1 RC oscilátor s Wienovým článkem

Základem tohoto oscilátoru je operační zesilovač s wienovým článkem v kladné zpětné vazbě. Schéma zapojení znázorňuje obrázek 2.1.



Obr. 2.1: Oscilátor s wienovým článkem

Oscilátor bude kmitat na frekvenci  $f_0$  na základě vztahu:

$$f_0 = \frac{1}{2\pi \cdot RC \cdot C} \quad [Hz][1] \quad (2.1)$$

Wienův článek je pásmová propust, která má při mezním kmitočtu  $f_0$  přenos roven  $1/3$ . Aby došlo ke vzniku oscilací, musí být splněna amplitudová podmínka, která říká, že přenos otevřené smyčky musí být větší než 1. Je tedy potřeba zajistit aby hodnota zesílení záporné zpětné vazby byla nejméně 3. Změnou této hodnoty lze také měnit amplitudu výstupního signálu[1].

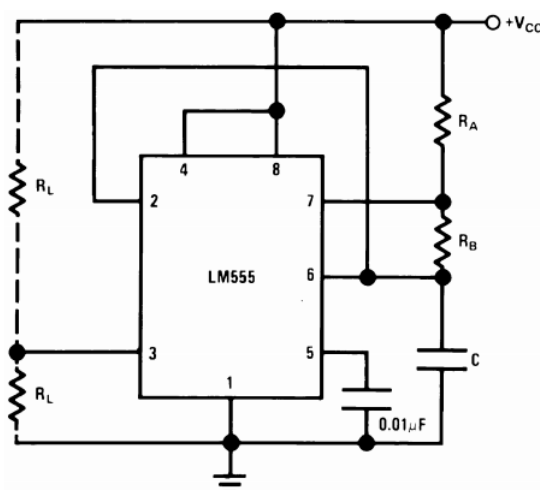
## 2.2 Integrované obvody

Jako generátor signálu lze také použít integrované obvody, kterých je na trhu spousta. Jedna z jejich hlavních výhod je realizace generátoru s malým počtem obvodových součástek. Použít lze obvody přímo určené pro generování signálů, kdy lze získat i více tvarů z jednoho obvodu, nebo použít některý z univerzálních obvodů kdy požadovaných vlastností lze docílit pomocí dalších obvodových součástek.

Z univerzálních obvodů lze zmínit asi nejslavnější časovací obvod 555. Z obvodů přímo určených pak ICL8038.

### 2.2.1 LM555

Jedná se o časovací obvod s celou řadou možností zapojení. Pro potřeby generování signálu lze použít zapojení jako astabilní klopný obvod, resp. multivibrátor. Schéma zapojení, viz obrázek 2.2



Obr. 2.2: Zapojení obvodu 555 jako multivibrátor

Po přivedení napájecího napětí se začne nabíjet kondenzátor C skrze rezistory  $R_A$  a  $R_B$  dokud napětí na C1 nepřesáhne  $2/3$  napájecího napětí, kdy se výstup obvodu sepne do log. 0. Poté obvod sepne pin č. 7 na zem a C se vybíjí přes  $R_A$  do doby, kdy napětí na C neklesne pod  $1/3$  napájecího a výstup je sepnut do log. 1. Poté následuje cyklus nabíjení kondenzátoru. Výstupní signál je tedy obdelníkového tvaru. Jeho frekvenci lze měnit změnou hodnot  $R_A, R_B$  a C na základě vztahu:

$$f = \frac{1.44}{(R_A + 2R_B) \cdot C} \quad [Hz] \quad (2.2)$$

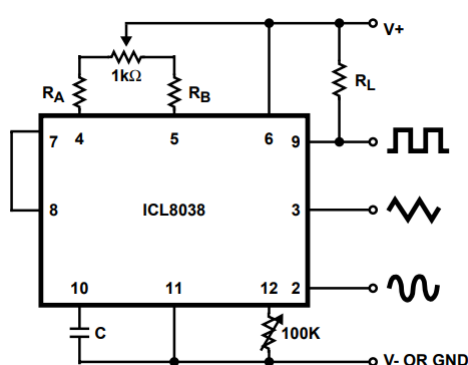


Ze zapojení lze usuzovat, že teplotní stabilita bude záviset na použitých součástkách  $R_A, R_B$  a  $C$ . Výrobce udává, při hodnotách součástek  $C=100\text{nF}$  a  $R_A=R_B$  v rozmezí 1 - 100kOhm, stabilitu  $\delta f = 150\text{ppm}/^\circ\text{C}$ [2].

## 2.2.2 ICL8038

ICL8038 od firmy Renesas je integrovaný obvod generátoru funkcí sinus, trojúhelník a obdelník navržený tak, že pro svůj chod potřebuje jen minimum externích součástek. Rozsah frekvencí je od 0,001Hz až po více než 300kHz. Obvod také umožňuje změnu střídy. Výrobce udává teplotní stabilitu  $\delta f = 250\text{ppm}/^\circ\text{C}$ [3].

Katalogové zapojení obvodu, viz obr. 2.3



Obr. 2.3: Katalogové zapojení obvodu[3]

## 2.3 Krystalové rezonátory

Krystal je elektronická součástka, která se v elektronice používá jako zdroj frekvence fungující na principu piezoelektrického jevu. To znamená, že pokud opatříme kus piezoelektrického materiálu elektrodami a budeme jej mechanicky deformovat, dojde k polarizaci piezoelektrika a na elektrodách se objeví elektrický náboj. U tohoto vzorku lze pozorovat také jev opačný. Tedy polarizaci piezoelektrika elektrickým polem se materiál deformuje. Jako materiál se nejčastěji používá oxid křemičitý  $\text{SiO}_2$ . Ten je ve formě krystalu nařezán na destičky, opatřené elektrodami a zapouzdřen[5][4].

Elektrické chování krystalu lze popsat pomocí jeho náhradního schématu, viz obr. 2.4 b). Lze si povšimnout, že se prakticky jedná o rezonanční obvod, který

může pracovat ve dvou oblastech, a to v sériové nebo v paralelní rezonanci. Při sériové rezonanci kmitá krystal na rezonanční frekvenci  $f_r$ :

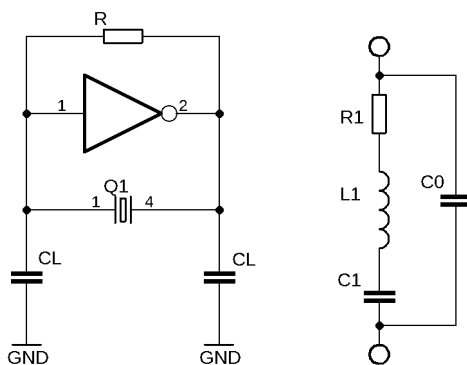
$$f_r = \frac{1}{2\pi\sqrt{LC}} \quad [Hz][4] \quad (2.3)$$

Při zvyšování frekvence se začne uplatňovat paralelní kapacita  $C_0$  a nastává paralelní rezonance:

$$f_a = \frac{1}{2\pi\sqrt{L\frac{C_1C_0}{C_1+C_0}}} \quad [Hz][4] \quad (2.4)$$

Výrobce v katalogovém listu udává pro který typ rezonance je krystal nalaďen. U paralelní rezonance bývá nalaďena při určité kapacitní zátěži  $C_L$ , jejíž hodnotu lze taktéž najít v katalogovém listu.

Jako oscilátor s použitím krystalu lze použít Piercovo zapojení s CMOS invertorem. Schéma zapojení, viz obr. 2.4 a).



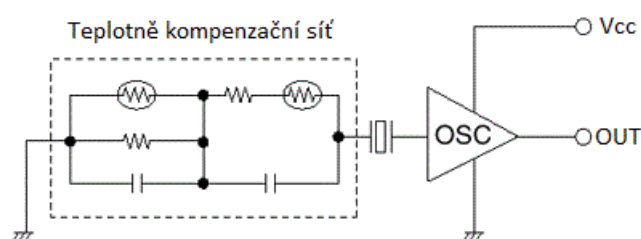
Obr. 2.4: a) Piercův oscilátor, b) Náhradní schéma krystalu

Jeden z parametrů ovlivňující frekvenční stabilitu krystalu je teplota. Běžné krystaly o frekvenci 1 - 10 MHz dosahují teplotní stability okolo 50ppm. Existují proto jejich modifikace, které jejich teplotní stabilitu značně zlepšují. Jedná se například o teplotně kompenzované krystaly (TCXO, Temperature Compensated Crystal Oscillator) nebo krystal s vyhřívanou komorou (OXCO, Oven Controlled Crystal Oscillator).

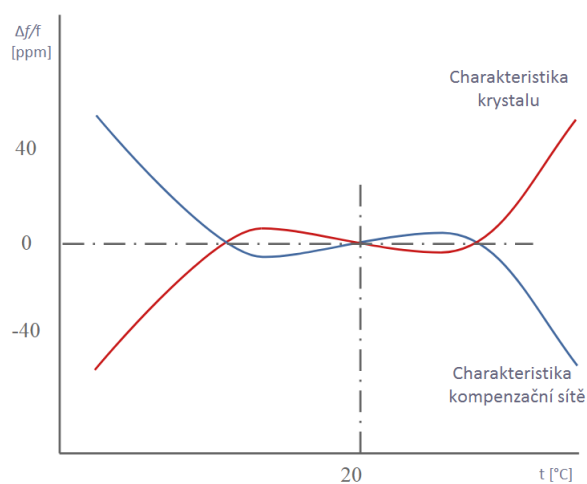
### 2.3.1 TCXO

Princip TCXO spočívá v zařazení krystalu do obvodu obsahující součástky s opačnou teplotní závislostí. Použít se dají například kapacitory nebo termistory. Obvod z těchto součástek je seskládán tak, aby bylo dosaženo co nejpodobnější teplotní charakteristiky krystalu, avšak s opačnou teplotní závislostí, jak znázorňuje obrázek 2.6. Jedno z možných řešení pak zobrazuje obrázek 2.5. Celý systém je pak zapouzdřen. Na zmiňovaném obrázku lze pozorovat, že systém obsahuje vlastní zpětnovazební obvod spolu s bufferem, takže k součástce stačí přivést napájecí napětí a na výstupu se objeví signál o parametrech daných výrobcem[5].

Takto kompenzovaný krystal pak dosahuje teplotní stability v mezích  $\delta f = 0,2 - 5 \text{ ppm}/^{\circ}\text{C}$



Obr. 2.5: Vnitřní struktura TCXO



Obr. 2.6: Kompenzace, graf[7].

### 2.3.2 OCXO

Princip spočívá v umístění krystalu do komory vytápěné na konstantní teplotu. Do komory je často umístěn celý obvod oscilátoru, tím se minimalizuje vliv teploty na jeho parametry. Prakticky je OCXO realizován jako kovová krabička uvnitř které se nachází samotná vytápěná komora, tepelně izolována od okolí. Součástí regulační smyčky vytápění je teplotní čidlo, které snímá teplotu pece. Na základě odečtené hodnoty pak akční člen udržuje stálou teplotu pomocí topného tělesa.

Tyto krystaly dosahují teplotní stability v rozsahu až  $\delta f = 0,001 - 0,1 \text{ ppm}/^{\circ}\text{C}$ . Obrázek 2.3.2 pak zobrazuje mechanické provedení krystalu[5].



Obr. 2.7: Zapouzdřené OCXO.

## 3 Generátor na bázi TCXO

Tato kapitola obsahuje teoretický rozbor, jehož výstupem je koncept pro generátor na bázi TCXO. Rozbor přibližuje v současné době dostupné programovatelné logické obvody a jejich možnost programování v jazyku VHDL. Dále pak přibližuje pojem digitálně analogový převodník. Na konci kapitoly je pak stanoven celkový koncept generátoru. Teplotně kompenzovaným krystalům se tato kapitola již nevěnuje. Teorie krystalů je probrána v kapitole 2.3.

### 3.1 Logické obvody

Logický obvod je elektronická součástka, která zpracovává elektrické signály v digitální podobě. Tento obvod se skládá z logických členů (hradel) které reprezentují logickou funkci, například logický součet nebo součin. Jako příklad lze uvést integrované obvody řady 74xx, které obsahují jeden či více logických členů.

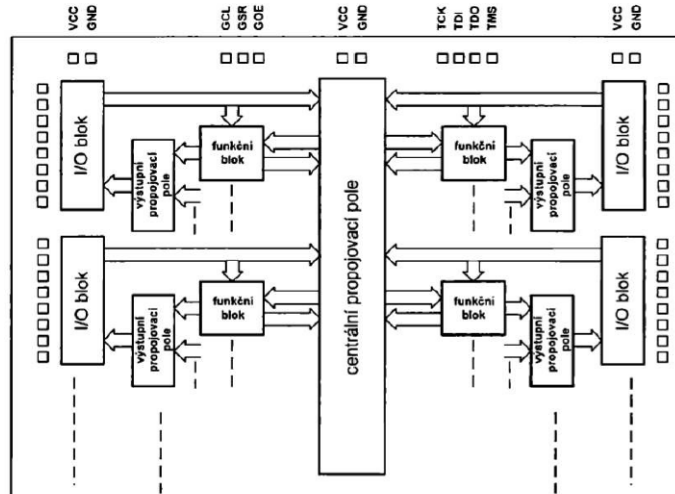
Při větším počtu logických obvodů nastává problém velkých rozměrů desek plošných spojů s vysokým počtem signálových cest. Pro odstranění těchto nevýhod byl vyvinut programovatelný logický obvod.

V dnešní době se nejčastěji používají komplexní programovatelné logické obvody (CPLD) a hradlová pole (FPGA). Pro jejich vnitřní hardwarový popis se využívá jazyků VHDL, Verilog a jiných.

#### CPLD

Jedná se o logický obvod menšího rozsahu. Hlavním blokem obvodu je makrobuňka, která obsahuje programovatelnou logiku pro vytváření logických a sekvenčních obvodů. Dnešní obvody obsahují 30 až 1 000 makrobuněk. Jednotlivé makrobuňky lze mezi sebou propojovat pomocí centrálního propojovacího pole. Propojení výstupu makrobuňky s vstupně výstupním blokem obstarává výstupní propojovací pole. Vnitřní strukturu obvodu znázorňuje obrázek 3.1.

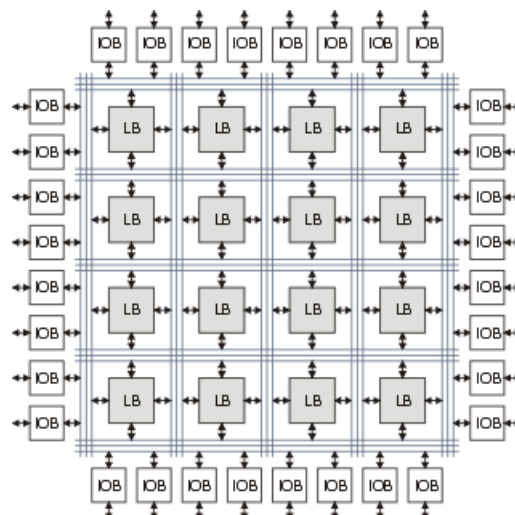
Výše uvedené bloky si uchovávají svoji konfiguraci pomocí buněk typu FLASH nebo EEPROM. Obvody si tedy pamatují svoji konfiguraci i po přerušení napájecího napětí. Nevýhodou těchto technologií je omezený počet přeprogramování (1 000 až 10 000 cyklů)[10].



Obr. 3.1: Vnitřní struktura CPLD

## FPGA

Jedná se o obecnější a mnohem rozsáhlejší obvody, než obvody typu CPLD. Základní buňkou FPGA je logický blok (Logic Block, LB) obsahující generátor logické funkce (tzv. LookUp Tlabe, LUT), a klopný obvod. Vzájemné propojení logických bloků a spojení se vstupně výstupními bloky zajišťuje vertikální a horizontální propojovací pole. Vnitřní strukturu obvodu znázorňuje obrázek 3.1. Obvody typu FPGA využívají pro uchování své konfigurace paměti typu RAM. Při startu je tedy nutno konfiguraci načíst z externí paměti[10].



Obr. 3.2: Vnitřní struktura obvodu FPGA

## 3.2 VHDL

VHDL (VHSIC HDL, Very High Speed Integrated Circuits Hardware Description Language ) je jazyk sloužící pro popis hardwaru. Používá se hlavně pro popis a simulaci číslicových obvodů. Jedná se o tzv. otevřený standart. To znamená, že pro jeho využití není potřeba licence, narozdíl od například ABEL-HDL. Z tohoto důvodu je stále hojně využíván. Další výhodou je využití kódu popisu hardwaru pro jeho následnou simulaci.[9]

### 3.2.1 Základní struktura jazyka VHDL

Konstrukce se dělí na dvě základní části a to sice deklaraci entity a popis architektury. Jako ukázka struktury je uveden model entity logického součinu dvou vektorů, viz výpis3.1.

Výpis 3.1: Struktura VHDL

<pre><u>library</u> IEEE; <u>use</u> IEEE.STD_LOGIC_1164.<u>ALL</u>;  <u>ENTITY</u> soucin <u>IS</u>   <u>PORT</u>(     a,b : <u>IN</u> STD_LOGIC_VECTOR (3 <u>DOWNTO</u> 0);     y : <u>OUT</u> STD_LOGIC_VECTOR (3 <u>DOWNTO</u> 0)   );   <u>END</u> soucin;    <u>ARCHITECTURE</u> Behavioral <u>OF</u> soucin <u>IS</u>   <u>BEGIN</u>     y &lt;= a <u>and</u> b;   <u>END</u> Behavioral;</pre>	<p>1 2 3 4 5 6 7 8 9 10 11 12 13 14</p>
--	---

V deklaraci entity jsou deklarovány tzv. brány v oblasti za klíčovým slovem *PORT*. Brána se skládá z identifikátoru (jména) signálu, jeho směru a datového typu. Směr signálu může být například typu *IN* pro vstupní signál, typ *OUT* pro výstupní signál nebo *INOUT* reprezentující obousměrný tok dat. Datových typů obsahuje jazyk VHDL celou řadu. Hojně používané typy jsou *STD\_LOGIC* a *STD\_LOGIC\_VECTOR*. Zde je také možnost volit směr vektoru. Například zápisem *n DOWNTO 0* vznikne vektor v binární podobě zleva doprava od bitu s největší vahou (MSB) po bit s nejmenší vahou (LSB). Při zápisu *0 TO n* je směr opačný, tedy od LSB po MSB.

Architektura může být popsána několika způsoby. Například behaviorální popis (výpis 3.1). Jedná o popis o popis chování hardwaru. Strukturální popis (výpis 3.2) slouží pro propojení jednotlivých uživatelských bloků (komponent).[10][9].

### Výpis 3.2: VHDL, strukturální popis

```

library IEEE;

ENTITY system IS
    PORT(
        A,B : IN STD_LOGIC_VECTOR (3 DOWNT0 0);
        Y : OUT STD_LOGIC_VECTOR (3 DOWNT0 0)
    );
END system;

ARCHITECTURE Structural OF system IS
BEGIN

    component
    PORT(
        a,b : IN STD_LOGIC_VECTOR (3 DOWNT0 0);
        y : OUT STD_LOGIC_VECTOR (3 DOWNT0 0)
    );
END component;

    logsoucín : soucín port map (A, B, Y);

END Structural;

```

Tělo architektury se skládá ze dvou částí. Deklarační a příkazové. Příkazová část obsahuje jednotlivé příkazy pro popis obvodu a je ohraničena klíčovým slovem `begin` a koncem architektury. Deklarační část je ohraničena začátkem architektury a klíčovým slovem `begin`. Deklarují se zde signály a konstanty.

Konstanta je neměnná hodnota, která se nejčastěji využívá pro přehlednost a efektivitu kódu. Její hodnota se přiřazuje při deklaraci, například:

*CONSTANT X: integer := 10;*

Signály jsou datové objekty, které tvoří realizovaný obvod, nejčastěji v podobě elektrických propojení. Při deklaraci jim může být přiřazená počáteční hodnota, viz: *signal tmp : STD\_LOGIC\_VECTOR (1 downto 0) := "11";*[9][10].

#### 3.2.2 Paralelní příkazy

Jedná se o příkazy, které se vykonávají souběžně. Nezáleží tedy na jejich pořadí ve zdrojovém textu. Jedním z nich je přiřazovací příkaz `<=`, nepodmíněné přiřazení. Příklad použití se nachází ve výpisu 3.3[10].



Výpis 3.3: Nepodmíněné přiřazení ve VHDL

```
y1 <= a and b ;
y2<= '0' or b ;
y3<= "001" and c ;
```

1  
2  
3

Na řádce č.1 se nachází realizace logického součinu, kdy signál y je výstup, a a b jsou vstupy. Na druhém řádce lze pozorovat, že lze pracovat i s konkrétní hodnotou, nikoliv jen se signály. Třetí řádek zobrazuje možnost práce se signály ve formě vektorů za předpokladu stejné šířky všech signálů.

## WITH-SELECT-WHEN

Příkaz with-select-when, neboli výběrové přiřazení hodnoty signálu na základě hodnoty jiného signálu. Tento příkaz se hojně využívá například pro realizaci dekodérů či multiplexoru, viz výpis 3.4.

Výpis 3.4: Čtyřbitový multiplexor ve VHDL

```
WITH s SELECT
  y <= a WHEN "00" ,
    b WHEN "01" ,
    c WHEN "10" ,
    d WHEN "11" ,
    '0' WHEN OTHERS ;
```

1  
2  
3  
4  
5  
6

Signál y je vstupní signál. a, b ,c a d jsou pak výstupní signály, všechny typu *std\_logic* . Signál s je adresový vstup typu *std\_logic\_vector* pomocí kterého se na výstupy přepínají vstupní signály. Klíčové slovo *others* zastává funkce všech ostatních kombinací signálu s, Tedy v tomto případě bude na výstupu logická nula.

### 3.2.3 Sekvenční příkazy

Jedná se o příkazy, které se vykonávají, podobně jakou u běžných programovacích jazyků, postupně. Jedním ze sekvenčních příkazu je příkaz if.

#### If

Využívá se pro vykonání příkazů na základě vyhodnocení podmínky. používá se uvnitř procesu, který na základě sensitivity listu, jenž se nachází v závorkách sa slovem process, spouští sekvenci příkazů. Používá se například pro tvorbu klopných obvodů, viz výpis 3.5[10].

Výpis 3.5: VHDL, klopný obvod typu D

```
process(clk)
```

```
begin
```

```
    if(rising_edge(clk)) then
```

```
        Q <= D;
```

```
    end if;
```

```
end process;
```

1

2

3

4

5

6

### 3.3 Digitálně analogový převodník

Digitálně analogový převodník (DA převodník) je elektronický obvod převádějící veličinu reprezentovanou v digitální formě a analogovou hodnotu veličiny. Tato veličina bývá ve formě proudu či napětí. Hodnota například napětového převodníku je dána vztahem:

$$U_o = D \cdot U_r \quad [V] \quad (3.1)$$

kde  $D$  je vstupní číslo převodníku v digitální podobě a  $U_r$  je hodnota referenčního napětí.

DA převodník se skládá ze zdroje referenčního napětí a obvodu skládajícího se ze sítě přesných rezistorů a elektronických spínačů ovládaných logikou převodníku.

Výstup převodníku bývá buďto napětový nebo proudový. Vstup může být realizován jako sériová sběrnice (například SPI nebo  $I^2C$ ) nebo jako paralelní sběrnice. Dále se vyrábějí převodníky s integrovaným zdrojem reference, nebo se vstupem pro externí referenci. Dále existují převodníky s různým výstupním rozlišením, například 8,10,12 nebo až 24 bitové.[12]

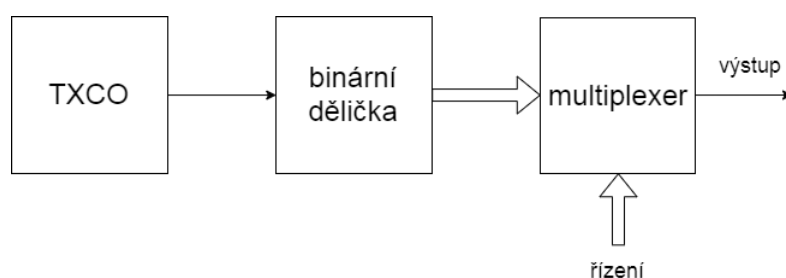
### 3.4 Návrh konceptu generátoru

Hlavní požadavek pro generátor je TCXO jako zdroj hodinového signálu a výstupní frekvence v rozmezí 1 – 20Hz obdelníkového signálu. Vedlejší, neformální, požadavky jsou možnost vícero hodnot výstupní frekvence spolu s paralelním výstupem trojúhelníkového signálu. Dále by se mělo jednat o jednodušší zařízení s použitím běžně dostupných součástek.

Nabízejí se dvě možnosti, jak zadání splnit. Buďto zvolit několik hodnot kmitočtu a pro jeho každou hodnotu a tvar signálu použít výstupní konektor, nebo použít pouze dva výstupy, pro každý signál zvlášť, a uživatelsky měnit hodnotu frekvence. První řešení odstraňuje problém přepínání kmitočtu, avšak za cenu vysokého počtu

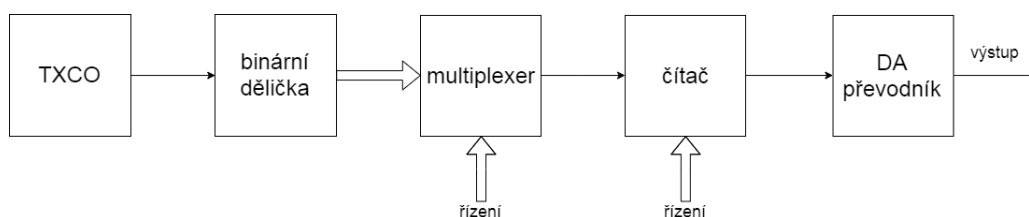
výstupních konektorů. Naproti tomu, druhé řešení obsahuje pouze dva konektory, ale zapojení musí obsahovat obvod pro změnu frekvence.

Na trhu se vyskytuje velká spousta TCXO s různou hodnotou výstupní frekvence v rozsahu 32768kHz (krystaly pro obvody reálného času, RTC) až po stovky MHz. S tím souvisí problematika modifikace signálu. Jednou z častých metod je použití digitální syntézy. Tato metoda je však poněkud složitějšího a náročnějšího charakteru. Jako jednodušší cesta se jeví použití krystalu o vhodném kmitočtu následně poděleným binární děličkou. Pokud je dělička doplněna o multiplexor, získá se tak výstup s binárně odstupňovanými hodnotami frekvence. Obrázek 3.3 nastiňuje možné blokové schéma zapojení.



Obr. 3.3: Blokové schéma binární děličky

Co se týká trojúhelníkového signálu, jedna z možností je použití obousměrného binárního čítače v kombinaci s digitálně analogovým převodníkem vybaveným paralelní vstupní sběrnicí. Zdroj hodinového signálu pro čítač zajišťuje předchozí zapojení, viz obrázek 3.4

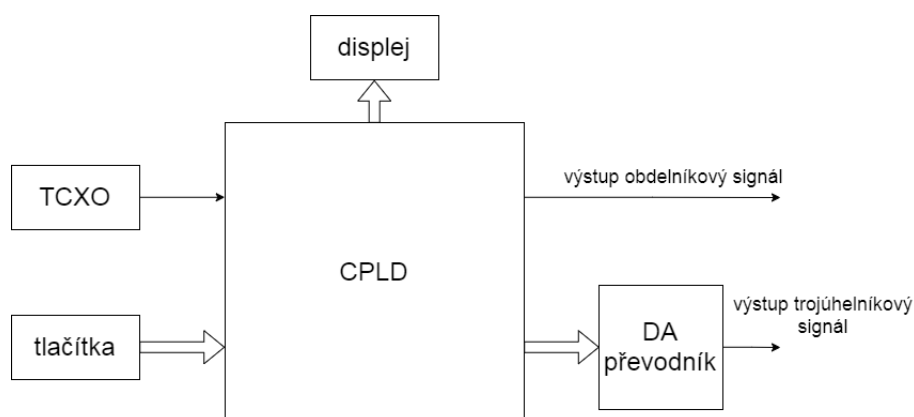


Obr. 3.4: Blokové schéma obvodu trojúhelníkového signálu

Kombinací těchto dvou zapojení lze dosáhnout obvodu, jehož výstupy jsou dva signály. Toto zapojení lze realizovat pomocí logických obvodů, například použitím obvodů řady 74xx. Zde je dále potřeba zajistit ovládání multiplexeru a čítače. To lze vyřešit například malým mikrokontrolérem spolu s displejem a tlačítky pro změnu frekvence. Další výhodou tohoto zapojení je možné doplnění obvodu o komunikační

obvod pro případnou parametrizaci generátoru pomocí měřicího systému, ke kterému bude generátor připojen.

Jak lze pozorovat, v návrhu mají majoritní převahu logické obvody. Vyskládání obvodu z jednotlivých součástek se poněkud neefektivní. Nabízí se možnost využití některého z programovatelných logických obvodů. Pro obvod menšího rozsahu, jako je tento je vhodné použití menšího obvodu typu CPLD. Ten dokáže zastat funkci veškeré modulace signálu, vyjma DA převodníku. Jedinými vstupními signály obvodu jsou hodinový signál z TCXO a signály ovládacích tlačítek pro změnu frekvence. Výstupy jsou pak obdelníkový signál, paralelní datová sběrnice pro DA převodník a sběrnice pro displej zobrazující výstupní frekvenci. Blokové schéma zapojení zobrazuje obrázek 3.5.



Obr. 3.5: Blokové schéma konceptu generátoru

## 4 Návrh generátoru

Tato kapitola se zabývá návrhem schématu obvodu generátoru hodinového signálu na bázi teplotně kompenzovaného krystalu s obdelníkovým a trojúhelníkovým výstupním signálem. Tyto výstupy jsou proudově posílené. Modifikace základního signálu z krystalu zajišťuje programovatelný logický obvod typu CPLD. Trojúhelníkový signál generuje, na základě logického obvodu, digitálně analogový převodník. Výstup je realizován dvojicí operačních zesilovačů zapojených jako sledovač. Hodnotu frekvence výstupních signálů lze měnit pomocí dvou tlačítek. Tyto frekvence jsou 1, 2, 4, 8, 16 a 32 Hertz zobrazující se na dvoumístném sedmisegmentovém displeji. Napájení zajišťuje konektor USB-B. Výstupní konektory jsou typu BNC.

Deska plošného spoje je realizována jako dvouvrstvá pro povrchovou montáž součástek. Její rozměry a rozložení je navrženo tak, aby jej bylo možné umístit do malé krabičky.

V podkapitolách jsou popsány jednotlivé části obvodu. Kompletní schéma generátoru se nachází v příloze A

### 4.1 Napájení a stabilizace

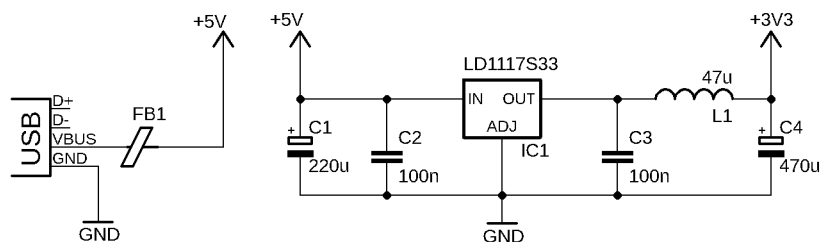
Napájení zajišťuje USB(Universal Serial Bus) konektor typu B. 5V větev je ke konektoru připojena pomocí feritové perly FB1. Ta potlačí rušení na vstupu. Zapojení, viz obrázek 4.1

Protože většina použitých obvodů pracuje s napájecím napětím 3,3V, je použit lineární stabilizátor s totožnou hodnotou výstupního napětí. Tolerance napětí USB sběrnice je 4,75 – 5,25V. Pro úbytek napětí na stabilizátoru tedy zbývá  $U = 4,75 - 3,3 = 1,45V$ . Běžné lineární stabilizátory pro svou správnou funkci potřebují úbytek aspoň 3V. Je zde tedy nutno použít LDO (Low-DropOut) stabilizátor. Je zvolen typ LD1117S33TR od firmy STMicroelectronics s parametry, viz tab. 4.1.

Tab. 4.1: Parametry stabilizátoru[13]

Max. vstupní napětí	$V_{in}$	15	V
Výstupní napětí	$V_{out}$	3.3	V
Max. výstupní proud	$I_o$	0.9	A
Úbytek na regulátoru	$V_d$	1.2	V

Z tabulky 4.1 vyplývá, že zvolený stabilizátor potřebuje minimální úbytek 1,2V. Tento typ tedy požadavky splňuje s rezervou 250mV. Na obrázku 4.1 se nachází schéma zapojení.



Obr. 4.1: Schéma zapojení: a) USB konektoru, b) stabilizátoru

C1, C2 a C3 jsou blokovací kondenzátory, C2 a C3 jsou umístěny co nejblíže pouzdru stabilizátoru. C4 a cívka L1 slouží jako LC filtr pro potlačení rušivých signálů. Tento obvod se v zapojení nachází dvakrát. Jeden obvod slouží k napájení digitálních obvodů, druhý pro obvody analogové. Každé zapojení má svou zem spojenou v jednom bodě.

## 4.2 TCXO

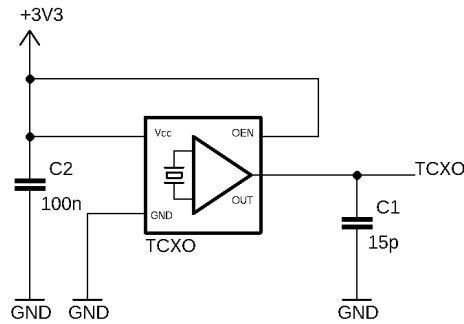
Jako zdroj hodinového signálu je zvolen teplotně kompenzovaný krystal ECS-327TXO od firmy ECS Inc. International. Katalogové hodnoty, viz tab. 4.2

Tab. 4.2: Tabulka parametrů krystalu[14]

Frekvence	32.768	kHz
Frekvenční stabilita	$\pm 1.5$	ppm
Teplotní stabilita frekvence	$\pm 5$	ppm
Časová stabilita frekvence	$\pm 3$	ppm
Napájecí napětí	1.8 – 5	V
Zatěžovací kapacita	15	pF

Hodnota frekvence 32,768kHz je zvolena kvůli snadnému podělení na hodnotu 1 Hz pomocí binární děličky. Dále nebude potřeba při návrhu DPS striktně dodržovat pravidla pro vysokofrekvenční obvody, narozdíl od krystalů s frekvencí v řádech desítek megahertz.

Krystal má výstup realizován jako CMOS, stejně jako vstupně výstupní piny logického obvodu. Je tedy možno krystal připojit přímo na vstup CPLD.



Obr. 4.2: Schéma zapojení TCXO

Schéma zapojení krystalu, viz obr. 4.2. Obvod je napájen napětím 3,3V. Aby byl výstup krystalu aktivní, je potřeba na vstup OEN přivést logickou úroveň 1, v tomto případě napájecí napětí. Blokovací kondenzátor C1 je umístěn co nejbližší pouzdra krystalu. Výrobce v katalogovém listu uvádí zatěžovací kapacitu 15pF, Na výstup je tedy připojen proti zemi kondenzátor C1 této hodnoty

## 4.3 CPLD

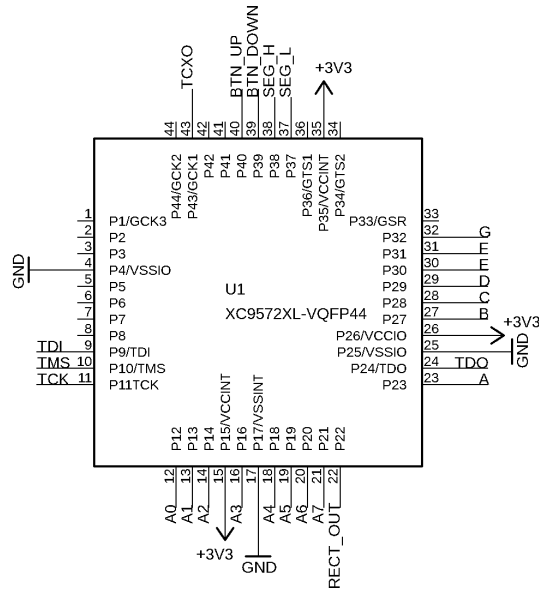
pro realizaci logických obvodů bloků děličky, obousměrného čítače a dalších bloků, zmíněných v kapitole 3.4 je vybrán programovatelný logický obvod XC9572XL, v pouzdře TQFP44, od společnosti Xilinx. Parametry důležité pro výběr obsahuje tabulka 4.3.

Tab. 4.3: XC9572XL, tabulka parametrů[15]

Počet makrobuňek	72
Max. pracovní frekvence	172MHz
Počet vstupně výstupních pinů	34
Napájecí napětí pro vnitřní logiku	3 – 3,3V
Napájecí napětí výstupních obvodů pro 3,3V logiku	3 – 3,6V

Obvod je srdcem celého generátoru. Do obvodu vstupuje hodinový signál TCXO z krystalu, který obvod upraví pro potřebu dalších obvodů na desce. Další vstupní signály jsou *BTN\_UP* a *BTN\_DOWN* přichází od ovládacích tlačítek. Mezi výstupní signály patří *RECT\_OUT*, který pokračuje do výstupního bufferu obdelníkového signálu. Osmibitová sběrnice *D0 – D7* vede k DA převodníku. K tomu vede také signál *WR*, jeho funkce je popsána v kapitole 4.6. Další výstupní signály *SEG\_H*, *SEG\_L*, *A*, *B*, *C*, *D*, *E*, *F*, *G* slouží k ovládání displeje. Aby bylo možné obvod

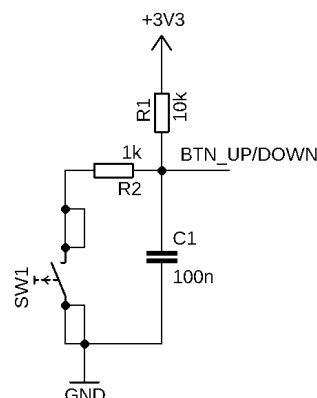
naprogramovat, jsou z obvodu vyvedeny signály pro JTAG sběrnici zakončené programovacím konektorem. U každého napájecího pinu je umístěna dvojice blokovacích kondenzátorů o hodnotách 0,1 a 1  $\mu\text{F}$ . Schéma zapojení obvodu a rozmístění signálů zobrazuje obrázek 4.3, chováním obvodu se pak zabývá kapitola 4.10



Obr. 4.3: Zapojení CPLD

## 4.4 Tlačítka

Pro změnu výstupní frekvence jsou použity mechanická tlačítka s obvodem pro omezení překmitů při stisknutí:



Obr. 4.4: Tlačítko s obvodem pro omezení zákmitů



Odpor R1 je tzv. pullup, který na vstupní pin přivádí napájecí napětí. V klidovém stavu je tedy na vstupu trvale napětí 3,3V. Přes tento odpor se také nabije kondenzátor C1. Při stisknutí tlačítka se kondenzátor začne vybíjet. Přes odpor R2 na hodnotu napětí danou děličem R1, R2.

## 4.5 Displej

Jako displej pro zobrazení výstupní frekvence slouží dvojice sedmisegmentových zobrazovačů SA39-12SRWA. Jedná se o jednodigitový displej červené barvy s velikostí znaku 9,9mm. Jednotlivé segmenty mají vyvedenou katodu zvlášť, anody pak spojeny všechny na jeden výstup, tedy zapojení se společnou anodou. Parametry jsou uvedeny v tabulce 4.4

Tab. 4.4: SA39-12SRWA, tabulka parametrů[16]

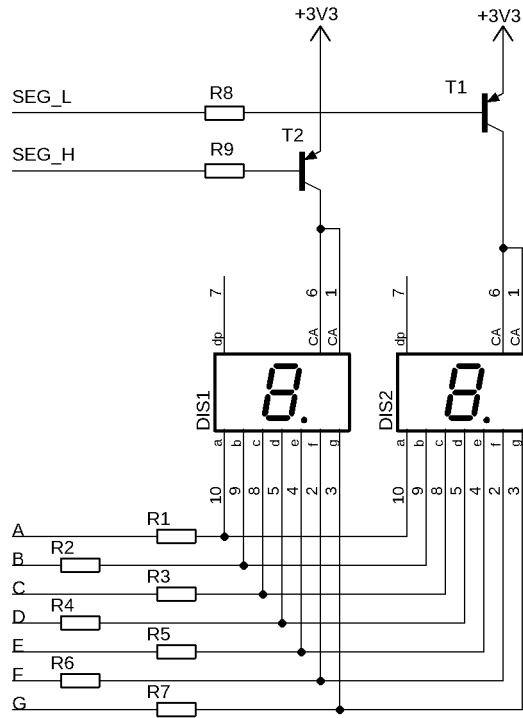
Napětí v propustném směru	$U_f$	1.8	V
Max. napětí v propustném směru	$U_{fmax}$	2.3	V
Napětí v závěrném směru	$U_r$	5	V
Proud v propustném směru	$I_f$	10	mA
Vlnová délka	$\lambda$	640	nm

Pro ovládání displeje je použito tzv. multiplexní řízení. To principiálně funguje tak, že je jedna datová sběrnice určující zobrazovaný znak. Pomocí druhé, adresové sběrnice je pak volen digit, na kterém má být znak zobrazen. Přepínání mezi digity pak probíhá na dostatečně vysoké frekvenci, kdy lidské oko není schopno postřehnout jednotlivé spínání digitů a zdá se, že svítí všechny digity současně.

Prakticky je toto zapojení realizované tak, že souhlasné katody segmentů obou displejů jsou spojeny a přes rezistor přivedeny na výstupní piny logického obvodu. Kombinace signálů A-G určuje zobrazovaný znak. Tranzistory T1 a T2 spínají na základě signálů *SEG\_L* a *SEG\_H* anody jednotlivých segmentovek. Jedná se o PNP tranzistory s parametry, viz tab. 4.5. Schéma zapojení zobrazuje obrázek 4.5.

Tab. 4.5: BC807, tabulka parametrů[17]

Max. proud kolektoru	$I_C$	0.5	A
Min. zesílení	$h_{FE}$	160	-
Max. napětí kolektor-emitor	$U_{CE}$	45	V
Sat. napětí kolektor-emitor	$U_{ce_{sat}}$	700	mV
Sat. napětí báze-emitor	$U_{be_{sat}}$	1200	mV



Obr. 4.5: Zapojení displeje

### Výpočet rezistorů R1-R7

Napájecí napětí je 3,3V. Napětí segmentu 1,8V při proudu 10mA. Na přechodu tranzistoru kolektor-emitor vzniká úbytek napětí 0,7V. Úbytek napětí na rezistoru musí být:

$$U_R = U_{CC} - V_f - U_{ce_{sat}} = 3.3 - 1.8 - 0.7 = 0.8V \quad (4.1)$$

Z proudu protékajícího rezistorem a požadovaném úbytku na něm lze zjistit jeho hodnotu pomocí Ohmova zákona.

$$R = \frac{U_R}{I_f} = \frac{0.8}{0.01} = 80\Omega \quad (4.2)$$

Nejbližší vyšší hodnota v řadě E24 je  $82\Omega$ . Je tedy zvolena tato hodnota.

### Výpočet rezistorů R8 a R9

Pokud je hodnota signálu  $SEG\_H/L$  rovna  $3,3V$ , rozdíl napětí mezi bází a emitorem je nulový, tranzistor je zavřený, tudíž žádný segment nemůže svítit. Změní-li se hodnota  $SEG\_H/L$  na  $0V$ , napětí mezi bází a emitorem nulové není a tranzistorem protéká proud.

V závislosti na zobrazovaném znaku se proud tranzistoru mění. V nejhorším případě bude sepnuto všech sedm segmentů, tedy tranzistorem poteče proud:

$$I_c = 7 \cdot I_f = 7 \cdot 0.01 = 70mA \quad (4.3)$$

Aby byl úbytek napětí na bázevých rezistorech dostatečný, je nutno počítat s minimálním zesílením tranzistoru, tedy  $h_{FE} = 160$ . Proud bází a požadovaný úbytek na rezistoru získáme podle následujících vztahů:

$$I_b = \frac{I_c}{h_{FE}} = \frac{0.07}{160} = 438\mu A \quad (4.4)$$

$$U_R = U_{CC} - U_{be_{sat}} = 3.3 - 1.2 = 2.1V \quad (4.5)$$

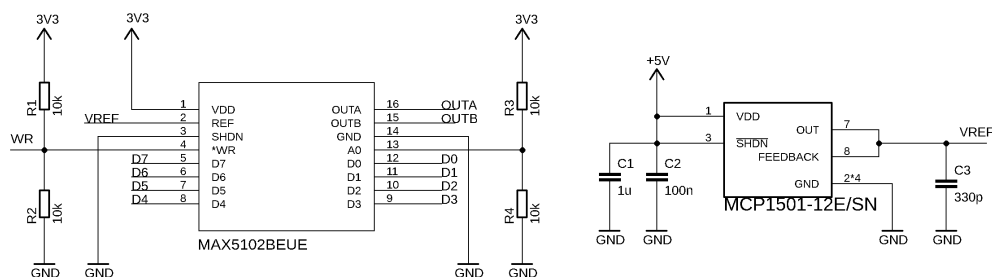
Hodnotu bázevého rezistoru lze lehce získat ze známého bázevého proudu a požadovaného úbytku napětí pomocí Ohmova zákona:

$$R = \frac{U_R}{I_b} = \frac{2.1}{0.000438} = 4895\Omega \quad (4.6)$$

Nejbližší vyšší hodnota v řadě E24 je  $5100\Omega$ . Volíme tedy tuto hodnotu.

## 4.6 DA převodník

Pro převod digitálního čísla na analogovou hodnotu je použit DA převodník MAX5102 od společnosti Maxim Integrated. Jedná se o dvojitý, 8bitový DA převodník s paralelní vstupní sběrnici. Vybrán byl pro jeho rozsah napájecího napětí ( $2,5 - 5,5V$ ) a vstupní paralelní sběrnici. Bohužel obvod neobsahuje referenční napětí, je tedy nutno použít externí. Jako reference je použit obvod MCP1501 s hodnotou napětí  $3,3V$  a přesností  $0,1\%$ . Zapojení obvodů znázorňuje obrázek 4.6



Obr. 4.6: Zapojení DA převodníku

Rezistory R3 a R4 slouží k výběru jednoho ze dvou výstupů převodníku. Je-li zapojen rezistor R4 a R3 je vynechán, je aktivní výstup A. V opačném případě je aktivní výstup B. Signál WR určuje zápis výstupu. Na první pohled není z data-sheetu jasné, zda vstup pracuje s hodnotou úrovně, či s hranou signálu. Signál WR je tedy vyveden do CPLD obvodu pro případné softwarové ovládání. Nastavením výstupu se pak zabývá kapitola 4.11.

Výstup z převodníku je ve formě schodovitého signálu, pro další vyhlazení signálu je za výstup převodníku použit RC filtr typu dolní propust. Hodnoty součástek jsou stanoveny na  $C=100\text{nF}$  a  $R=2200\Omega$ . To odpovídá mezní frekvenci:

$$f_0 = \frac{1}{2\pi \cdot R \cdot C} = \frac{1}{2\pi \cdot 2200 \cdot 100 \cdot 10^{-9}} = 796\text{Hz} \quad (4.7)$$

připojení výstupu převodníku k výstupnímu bufferu se provede připojením rezistoru filtru na patřičné místo. Pro výstup A se zapojí rezistor R19 (viz příloha A), pro výstup B se zapojí na místo R18. Konfigurace je popsána v kapitole 4.11.

## 4.7 Výstupní buffer

Výstupní buffer je realizován pomocí dvojice operačních zesilovačů (OZ), zapojených jako sledovač, pro každý výstup zvlášť. Použity jsou zesilovače MCP6021 od firmy Microchip s parametry uvedenými v tabulce 4.6.

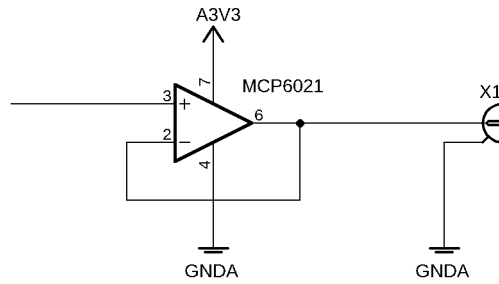
Jedná se o zesilovač s rail-to-rail vstupy a výstupem. To znamená, že OZ je schopen pracovat s hodnotami signálů velmi blízkým hodnotě napájecího napětí. Jelikož OZ pracuje se signály o maximální hodnotě 3,3V, a jeho napájecí napětí je totožné hodnoty, je zvolen právě tento typ zesilovače. Pro tento model udává výrobce saturací napětí 20mV od horní hranice napájecího napětí. Dále je schopen do zátěže dodat proud až 30mA. To odpovídá impedanci zátěže zhruba  $100\Omega$ . Vstupní impedance měřicího systému, ke kterému bude generátor připojen se pohybuje v řádech

Tab. 4.6: MCP6021, parametry[18]

Napájecí napětí	$V_{DD}$	2,5 – 5,5	V
Šířka přenášeného pásma	$GBWP$	10	MHz
Saturační napětí	$V_{OL}, V_{OH}$	-15, +20	mV
Výstupní proud nakrátko	$I_{SC}$	30	mA
Vstupní impedance	$Z$	$10^{13}$	$\Omega$

kilo ohmů. Hodnota výstupního proudu je tedy dostatečná. Vstupní odpor  $10^{13}\Omega$  je dostatečně velký, aby neovlivňoval RC filtr na výstupu DA převodníku.

Výstupní signál z bufferu je pak přiveden na výstupní konektor typu BNC. Schéma zapojení, viz obr 4.7.



Obr. 4.7: Zapojení výstupního bufferu

## 4.8 Návrh DPS

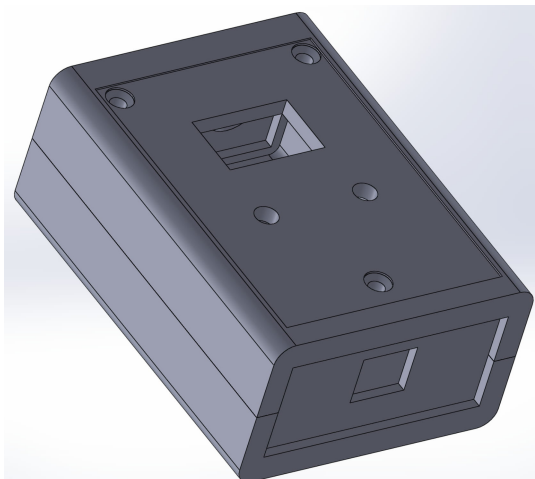
Deska plošných spojů je realizována jako dvouvrstvá, standardní tloušťky 1.6mm. Její rozměry jsou 51mm šířky a 71mm výšky. Deska je uzpůsobena pro povrchovou montáž a součástky jsou rozloženy tak, aby ji bylo možné umístit do krabičky. Rozložení cest a osazovací schéma se nachází v příloze B

Ve vrstvě Top se nacházejí ovládací tlačítka a displej, které prochází skrz otvory ven z krabičky. Vrstva bottom pak obsahuje součástky s vysokým profilem, které by se nevešly ve vrstvě top mezi desku a tělo krabičky, tedy programovací konektor pro JTAG a konektory BNC spolu s USB. Dále se zde nachází také elektrolytické kondenzátory. Již zmiňované konektory jsou rozmístěny u krajů tak, aby mohly procházet skrz čela krabičky ven.

Ostatní součástky, jako rezistory, kondenzátory, integrované obvody a ostatní jsou pak vhodně rozmístěny po obou stranách desky.

## 4.9 Krabička

Pro DPS je vymodelována malá kompaktní krabička o rozměrech 82mm výška, 60mm šířka a 36mm hloubka, která se skládá ze čtyř částí. Ty pak jako celek tvoří krabičku, viz obr. 4.8.



Obr. 4.8: Sestava krabičky

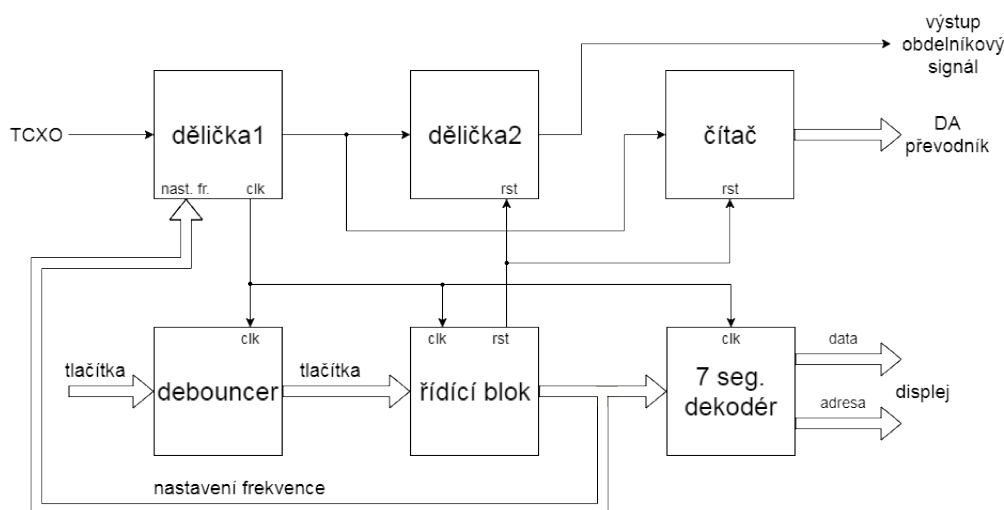
Vrchní část obsahuje otvory pro displej a tlačítka. Z vrchní strany je vytvořeno vybrání pro štítek, který je vyroben na zakázku. Zámky po stranách zajišťují upevnění čela krabičky. Z vnitřní strany je pak vymodelováno osazení pro DPS spolu s montážními otvory pro její uchycení a následnému sešroubování celé krabičky. Spodní část je tvarově prakticky identická s vrchní. Jsou v ní opět otvory pro šrouby, pouze spolu s osazením pro DPS chybí otvory pro tlačítka a displej.

Boční čela jsou vymodelovány tak, aby přesně zapadly do zámků předchozích dvou dílů. V těchto čelech se pak nachází otvory pro konektory USB na jedné, a BNC na druhé bočnici. Příloha D zobrazuje jednotlivé části krabičky.

## 4.10 Programové řešení

Samotný popis chování obvodu CPLD je napsán v jazyce VHDL. Program je koncipován do šesti jednotlivých funkčních bloků, takzvaných komponentů, z nichž každý zastává svou specifickou funkci. Tyto komponenty jsou navzájem propojeny pomocí signálových cest, jak naznačuje obrázek 4.9.

Celý popis vytváří tzv. hierarchický model, kdy jsou jednotlivé komponenty pospojovány do jednoho celku pomocí strukturálního popisu. Dílčí komponenty jsou pak popsány behaviorálním stylem. Tento způsob popisu celkově zpřehledňuje kód, a tím usnadňuje odhalení a následné odstranění případných chyb. Samotný strukturální popis se nachází v příloze F.1.



Obr. 4.9: Blokové schéma, funkční popis programu

## Dělička1

Tato komponenta slouží k prvotnímu podělení frekvence. Hodnota vektoru *output* je při každé náběžné hraně signálu z TCXO inkrementována o jedničku. Děje se tak do maxima, kdy hodnota osmibitového vektoru přeteče a čítá zase od začátku. Hodnoty Jednotlivých bitů signálu jsou pak posílány ven z komponenty signálem *main\_divider\_clk\_out\_signal* na základě hodnoty vektoru *main\_divider\_value*. Tento signál vstupuje do děličky2 a zároveň do obousměrného čítače. Uvedený popis realizuje osmibitovou binární děličku spolu s multiplexerem. Jeden z výstupů komponenty je taky signál *main\_divider\_clk\_out\_core*. Jedná se o vnitřní hodinový signál pro sedmisegmentový dekodér, řídicí obvod a debouncer. Jeho význam je popsán u jednotlivých komponentů. Popis komponenty obsahuje příloha F.2.

## Blok ovládání

Blok ovládání zajišťuje změnu frekvence na základě signálu od debounceru. Při spuštění generátoru je implicitně nastavena hodnota 1Hz. Komponenta funguje tak, že

při minimální frekvenci nereaguje na signál pro snížení frekvence a naopak. Při validnímu vyhodnocení změny frekvence komponenta vyše s příští náběžnou hranou vnitřního hodinového signálu reset signál, čímž vynuluje čítač a děličku2. tyto komponenty jsou nulovány aby nedošlo k fázovému posunu mezi výstupy generátoru. Popis komponenty se nachází v příloze F.6.

## **Dělička2**

Tato komponenta se stará o další podělení signálu z předchozí děličky a posílá jej ven z obvodu jako výstup obdelníkového signálu. Princip této děličky je totožný s předchozí. Hodnota  $n = 512$  kterou dělička dělí je neměnná, proto zde chybí multiplexer. Význam této hodnoty je popsán u komponenty čítače. Hodnota děličky je nulována signálem *divider\_rst*. Důvod nulování je popsán u komponenty blok ovládání. Samotný kód komponenty se nachází v příloze F.3.

## **Obousměrný čítač**

Na základě signálu z děličky1 čítač čítá nahoru a dolů. Hodnota čítače je posílána ven z logického obvodu do DA převodníku v podobě osmibitového vektoru. Jde o hodnotu okamžitého výstupního napětí generátoru v číslicové formě. DA převodník je osmibitový, za jednu periodu výstupního signálu tedy převodník musí převést 512 hodnot. Výstup z děličky2 je zároveň obdelníkovým výstupem generátoru. Aby byly frekvence obou výstupů rovny, musí čítač pracovat s vstupním signálem, jehož frekvence je 512krát rychlejší, než obdelníkový signál. Proto je výstupní obdelníkový signál oproti vstupu do čítače podělen, pomocí děličky2, hodnotou  $n=512$ . Popis komponenty čítače se nachází v příloze F.5.

## **Dekodér pro 7 segmentový displej**

Tato komponenta zajišťuje zobrazení výstupní frekvence generátoru na displeji na základě vstupního vektoru vycházejícího z bloku řízení. Displej je zapojen multiplexně, proto má komponenta dva výstupy. Jsou jimi datový a adresový vektor. Pomocí datového vektoru jsou zobrazovány znaky na displeji, datový vektor pak přepíná mezi jednotlivými digity o frekvenci vnitřního hodinového signálu. Popis komponenty se nachází v příloze F.7

## **Debouncer**

Debouncer odstraňuje zákmity vznikající při sepnutí, či rozepnutí mechanického tlačítka. Tato komponenta pracuje tak, že je-li detekována sestupná hrana vstupního signálu, jsou zaznamenány tři po sobě jdoucí vzorky se vzorkovací frekvencí totožnou

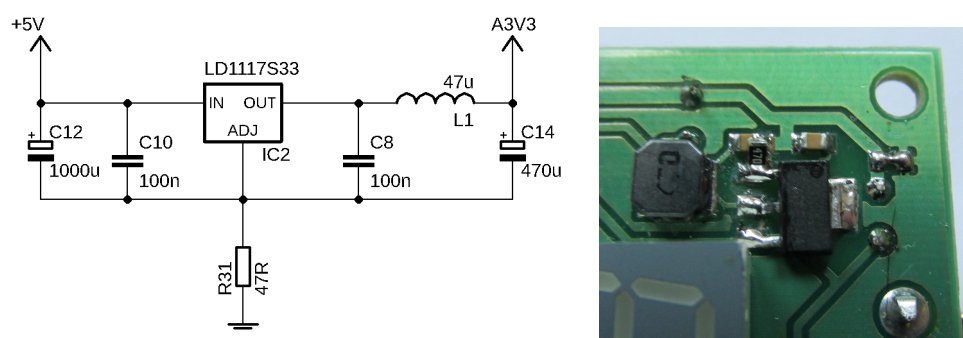


s vnitřním hodinovým signálem. Pokud jsou všechny hodnoty rovny nule, komponenta výsledek vyhodnotí jako stisknutí tlačítka a předá informaci řídicímu bloku. Popis komponenty obsahuje příloha F.4.

## 4.11 Osazení a oživení

Po osazení DPS součástkami je potřeba experimentálně určit frekvenci pro řízení displeje a debounceru. Jako nejprůběhovější se jeví hodnota 128Hz jak z hlediska displeje, tak odezvy tlačítek. Tato hodnota je upravena v děličce, viz kapitola 4.10.

Dále se přistoupilo k měření parametrů. Po připojení zátěže ke generátoru se pomocí osciloskopu objevila anomálie, kdy se měnila výstupní amplituda se změnou zátěže. Tento problém je zapříčiněn nedostatečným prostudováním katalogového listu operačního zesilovače. Výrobce udává saturační napětí 20mV, avšak tato hodnota silně závisí na výstupním proudu. Tento jev je odstraněn zvýšením napájecího napětí posunutím nuly stabilizátoru tak, že je na DPS přerušena cesta země stabilizátoru IC2 (viz příloha A) a přemostěna rezistorem. Schéma úpravy a její realizaci zobrazuje obrázek 4.11.



Obr. 4.10: Schéma a realizace úpravy napájecího napětí zesilovačů

Jak je zmíněno v kapitole 4.1, u stabilizátoru je napěťová rezerva 250mV. Napětí tedy lze tedy zvýšit maximálně o tuto hodnotu. Experimentálně je zjištěna hodnota rezistoru  $R31 = 47\Omega$ , kdy se napětí stabilizátoru zvýšilo o 210mV na hodnotu 3,51V.

Po této úpravě však klesla maximální hodnota výstupního proudu na 10mA, při které ještě nedochází k ořezání signálu.

u DA převodníku je zvolen výstup B. Osazeny jsou tedy rezistory R15 a R18. R14 a R19 jsou vynechány (v rozpisce materiálu nejsou uvedeny). Převodník zpracovává signál WR jako hodnotu hladiny signálu. Softwarové ovládání pomocí obvodu

CPLD tedy není třeba. Proto je signál WR uzemněn pomocí rezistoru R16. R17 není zapojen (v rozpisce materiálu není uveden).

příloha E zobrazuje osazenou DPS a sestavený generátor.

## 4.12 Měření parametrů

Po úspěšném oživení následuje měření parametrů generátoru. Měřeny jsou následující parametry:

### Amplituda signálu

Hodnota amplitudy je odečtena na osciloskopu OWON DS6062V za pomoci kurzorů. Měření probíhalo při zátěži  $500\Omega$ . Naměřené hodnoty obsahuje tabulka 4.7

Tab. 4.7: Měření amplitudy signálů, tabulka naměřených hodnot

signál	hodnota	jednotka
Obdelník	3,28	V
trojúhelník	3,27	V

### Náběžná a sestupná hrana obdelníkového signálu

Pro měření hran signálu je opět použit osciloskop OWON DS6062V s využitím kurzorů. Čas náběžné i sestupné hrany je měřen v intervalu 10-90% amplitudy při frekvenci 1Hz a zátěži  $500\Omega$ . Naměřené hodnoty, viz tabulka 4.8.

Tab. 4.8: Měření hran obdelníkového signálu, tabulka naměřených hodnot

hrana	hodnota	jednotka
náběžná	428	ns
sestupná	340	ns

### Frekvenční stabilita

Pro měření dlouhodobé frekvenční stability je zapotřebí měřicí systém, který periodicky měří frekvenci po dobu několika dnů. Takový systém však nebyl poskytnut

a proto toto měření není realizováno. Namísto toho je provedeno měření teplotní stability frekvence.

Průběh měření je následovný: Nejprve je generátor zapnut na 24 hodin aby se zahřál na provozní teplotu. Po zahřátí je změřena jeho frekvence. Po změření je generátor na 2 hodiny vypnut aby vychladl. Po vychladnutí je opět zapnut a změřeny parametry. Měření musí proběhnout co nejrychleji, aby se omezil vliv zahřívání.

Měření studeného a zahřátého přístroje nedává smysl. Před samotným použitím přístroje se vždy nechává zapnut po dobu aspoň 10 minut právě z důvodu, aby se ustálila jeho teplota. Tento způsob měření byl zvolen proto, aby se zjistila případná teplotní závislost výstupu. Za běžných provozních podmínek by nemělo dojít k větším výkyvům teplot v měřící laboratoři.

Měření je realizováno pomocí čítače HP53131A v režimu měření periody s následujícím nastavením: Vstupní impedance  $1M\Omega$ , průměrování 10 vzorků, spouštěcí úroveň 1V. Měření je provedeno pro oba výstupy a všechny frekvence 1 a 16 Hz. Naměřené hodnoty se nachází v tabulce 4.9.

Při obdelníkovém signálu, frekvenci 1 Hz a zahřátém generátoru je odchylka od jmenovité frekvence  $4,37\mu\text{Hz}$ . Rozdíl frekvencí při zahřátém a studeném generátoru je  $1,43\mu\text{Hz}$ . U trojúhelníkového signálu je tento rozdíl  $6,29\mu\text{Hz}$ . Při frekvenci 16Hz narostla tato hodnota o řád u obdelníku a dokonce o 2 řády u trojúhelníku.

Tab. 4.9: Měření periody signálů

obdelník					
	zatepla		zastudena		
f[Hz]	T[s]	f[Hz]	T[s]	f[Hz]	$ \delta f [\mu\text{Hz}]$
1	1 000 004 367	0,999 995	1,000 003 205	0,999 997	1,43
16	0,062 500 273	15,999 930	0,062 500 624	15,999840	89
trojúhelník					
1	1,000 010 560	0,999 989	1,000 004 270	0,999 996	6,29
16	0,062 498 385	16,000 413	0,062 500 260	15,999 993	480

Příklad výpočtu hodnot pro první řádek tabulky:

$$f = \frac{1}{T} = \frac{1}{1,000004367} = 0,999995\text{Hz} \quad (4.8)$$

$$|\delta f| = \left| \frac{1}{T_{tep}} - \frac{1}{T_{stud}} \right| = \left| \frac{1}{1,000004367} - \frac{1}{1,000003205} \right| = 1,43\mu\text{Hz} \quad (4.9)$$

## Závěr

V teoretické části práce jsou uvedeny základní parametry generátorů. Dále je zde uvedeno několik zdrojů hodinového signálu, včetně krystalových oscilátorů spolu s možnostmi jejich teplotní stabilizace.

V práci je také uveden teoretický rozbor na základě kterého je stanoven koncept generátoru na bázi teplotně kompenzovaného krystalu.

Praktická část práce se zabývá návrhem obvodu generátoru. V jednotlivých podkapitolách se nachází návrh dílčích částí obvodu, včetně výpočtu hodnot součástek.

Při oživování je objevena chyba návrhu, kdy je stanovena malá hodnota napájecího napětí operačních zesilovačů. Chyba je odstraněna zvýšením napájecího napětí. Nakonec se povedlo generátor dostat do funkčního stavu. Pro zařízení je také vymodelována malá krabička. Konečný vzhled generátoru se nachází v příloze.

Po úspěšném oživení je provedeno měření parametrů. Měření jsou následující parametry: Výstupní amplituda obou signálů. Náběžná a sestupná hrana obdelníkového signálu, kdy náběžná hrana dosahuje rychlosti 428ns a sestupná 340ns. Dále je pak měřena přesnost a stabilita frekvence. Odchylka výstupní frekvence 1Hz obdelníkového signálu činí  $1,43\mu\text{Hz}$ . U hodnoty 16Hz je to pak  $69,9\mu\text{Hz}$ .

# Literatura

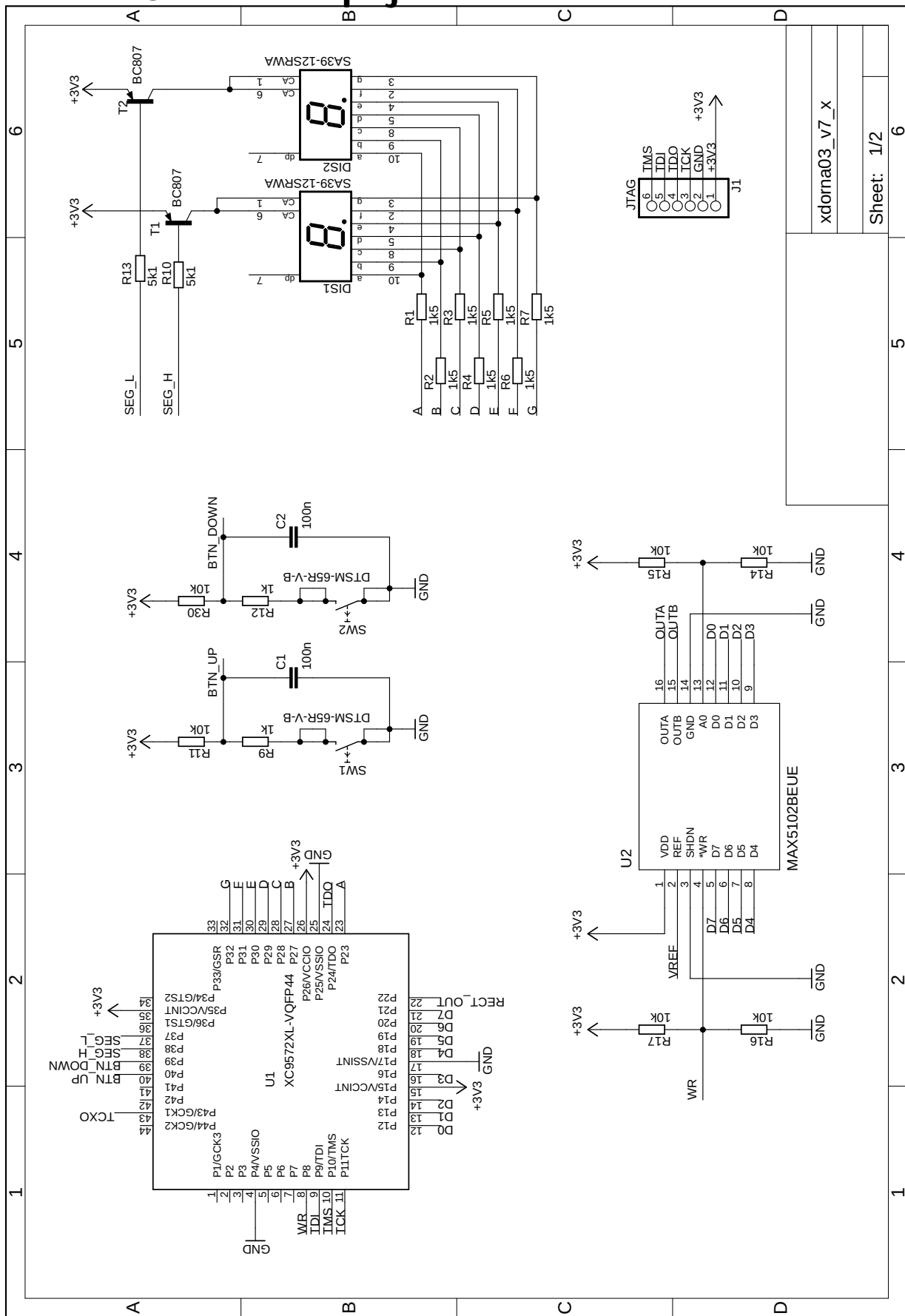
- [1] RADIM, Vít. *Funkční generátor. Purkyňova 97. Střední průmyslová škola elektrotechnická a informačních technologií, Brno, 2014*
- [2] Texas Instruments. *LM555 Timer. Datasheet, vydání SNAS548D, únor 2000.*
- [3] Intersil. *ICL8038, Precision Waveform Generator/Voltage Controlled Oscillator. Datasheet, vydání 2864.3, říjen 1998.*
- [4] VRBA, Kamil. *Konstrukce elektronických zařízení, Výběr a aplikační pravidla pro pasivní elektronické prvky, Vysoké učení technické v Brně, 2019 .*
- [5] Hewlett Packard. *Fundamentals of Quartz Oscillators, Application Note 200-2, 1997..*
- [6] *About Oscillator, Epson* [online]. [cit. 13. 05. 2020]. Dostupné z URL:   
<[https://www5.epsondevice.com/en/information/technical\\_info/osc/#2](https://www5.epsondevice.com/en/information/technical_info/osc/#2)>.
- [7] *TCXO, Temperature Compensated Crystal Oscillator, Electronic notes* [online]. [cit. 13. 05. 2020]. Dostupné z URL:   
<[https://www5.epsondevice.com/en/information/technical\\_info/osc/#2](https://www5.epsondevice.com/en/information/technical_info/osc/#2)>.
- [8] *LGH300 Ultra-Low-G-Sensitivity OCXO, Esterline Research & Design* [online]. [cit. 13. 05. 2020]. Dostupné z URL:   
<<https://www.esterlineresearch.com/products/lgh300-ultra-low-g-sensitivity-ocxo>>.
- [9] KOLOUCH, Jaromír. *Jazyk VHDL a jeho užití pro syntézu číslicových systémů. Brno, 2005.*
- [10] PINKER, Jiří a Martin POUPA. *Číslicové systémy a jazyk VHDL. 1. Věšínova 5, Praha 10: BEN - technická literatura, 2006. ISBN 80-7300-198-5.*
- [11] *Programovatelné logické obvody* [online]. [cit. 30. 04. 2020]. Dostupné z URL:   
<<http://fpga.sweb.cz/>>.
- [12] HAASZ, Vladimír a Miloš SEDLÁČEK. *Elektrická měření: přístroje a metody. 1. Praha: České vysoké učení technické, 1998. ISBN 80-010-1717-6.*
- [13] ST Microelectronic. *LD1117, Adjustable and fixed low drop positive voltage regulator. Datasheet, vydání 37, 2020*

- [14] ECS, Inc. International. *ECS-327TXO 32.768 KHz SMD TCXO. Datasheet, 2019*
- [15] Xilinx. *XC9572XL High Performance CPLD*. Product Specification. Datasheet, vydání DS057 (v2.0), duben 2007.
- [16] Kingbright, *SA39-12SRWA, Single Digit Numeric Display* . Datasheet, vydání 1A, červenec 2018.
- [17] Diodes Incorporated. *BC807-16/-25/-40 45V PNP small signal transistor in SOT23. Datasheet, vydání 19-2, říjen 2013.*
- [18] Microchip technology Inc. *MCP6021/1R/2/3/4, Rail-to-Rail Input/Output, 10 MHz Op Amps. Datasheet, vydání DS20001685E.*

# Seznam příloh

A Schéma zapojení	47
B DPS a osazovací schéma	49
C Rozpiska materiálu	50
D Části modelu krabičky	51
E Foto, osazená DPS a oživení	52
F Zdrojové soubory VHDL	53
G Obsah CD	63

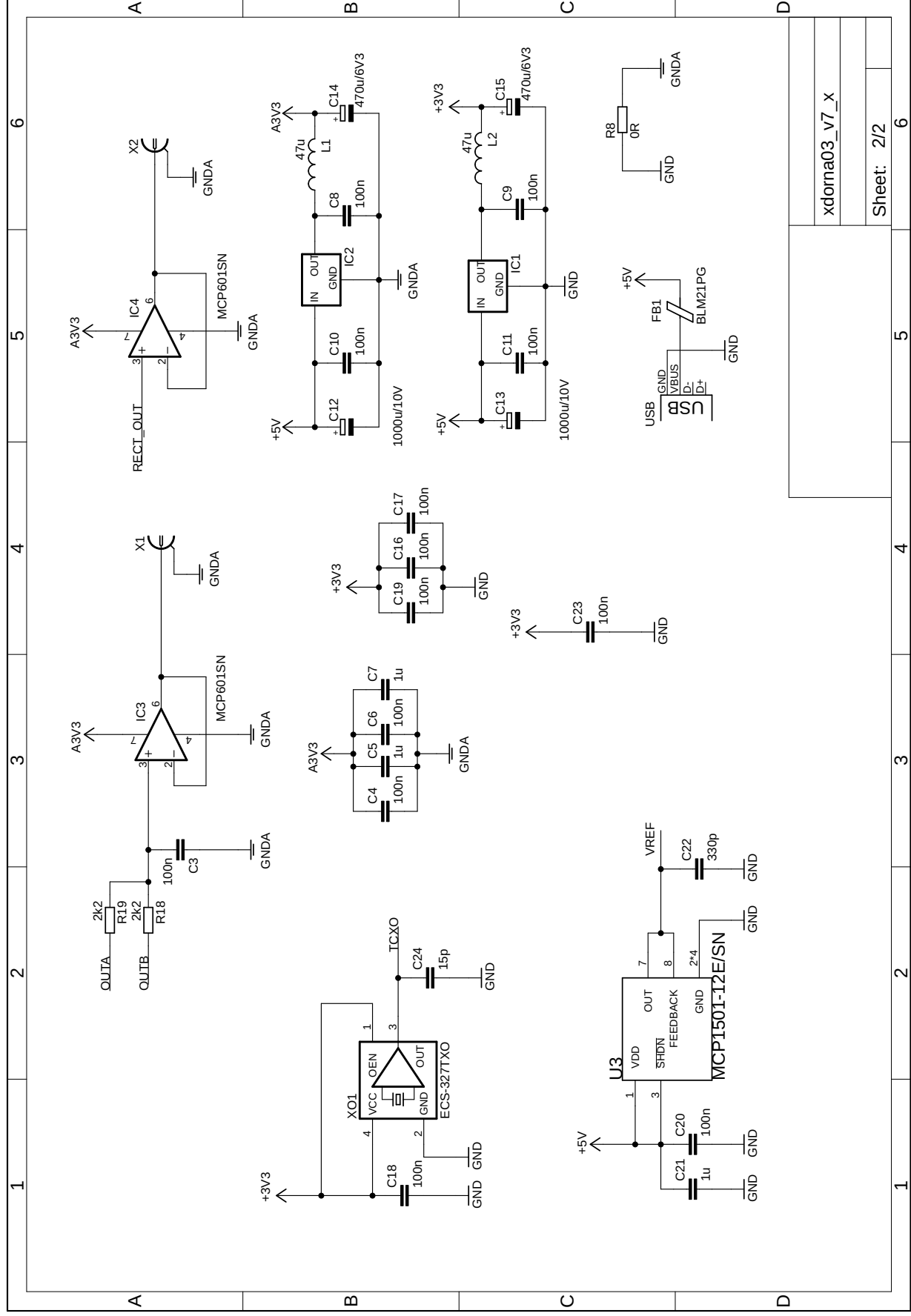
# A Schéma zapojení



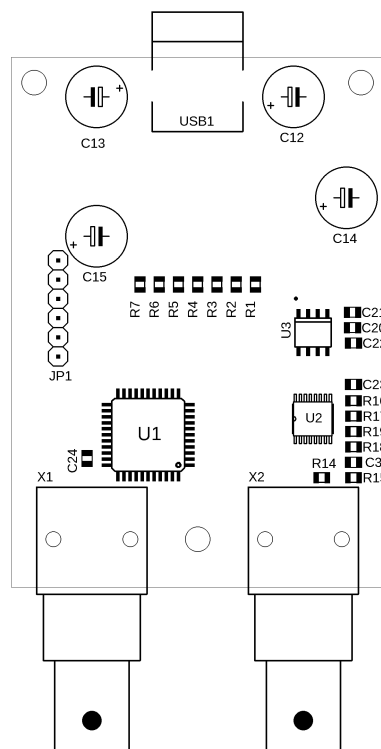
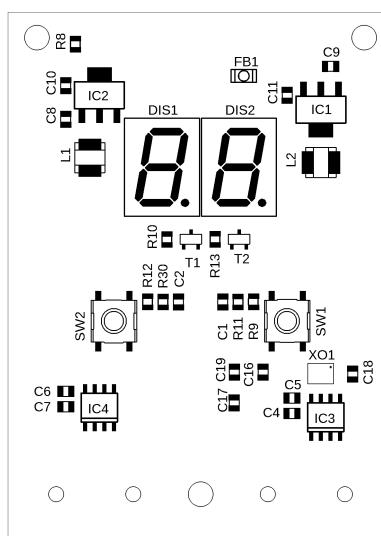
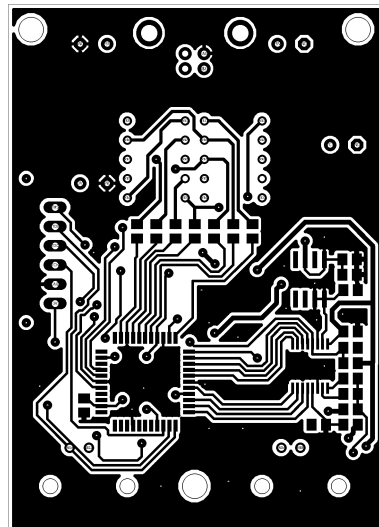
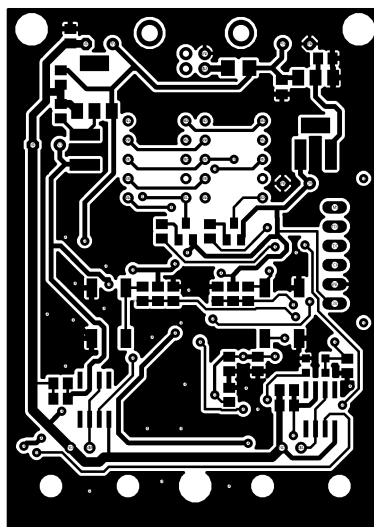
xdorna03\_v7\_x

Sheet: 1/2





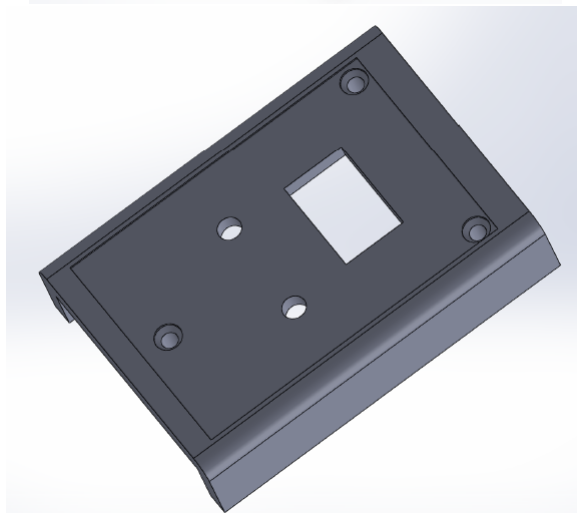
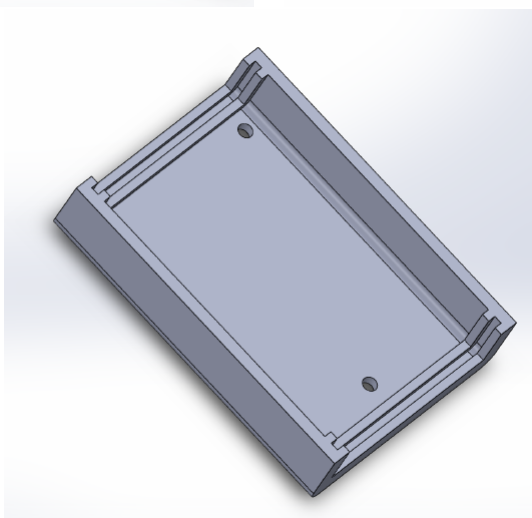
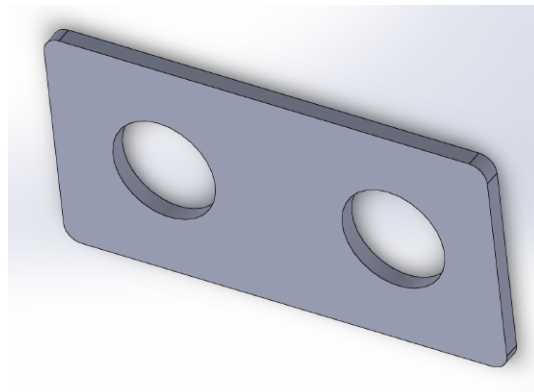
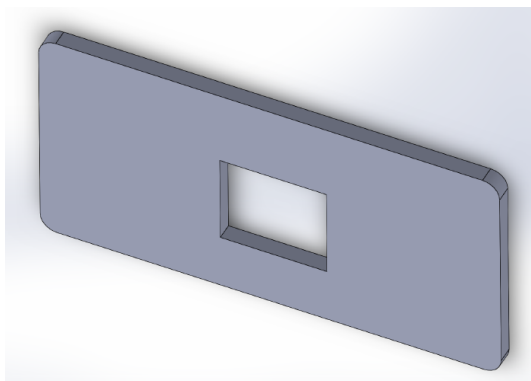
## B DPS a osazovací schéma



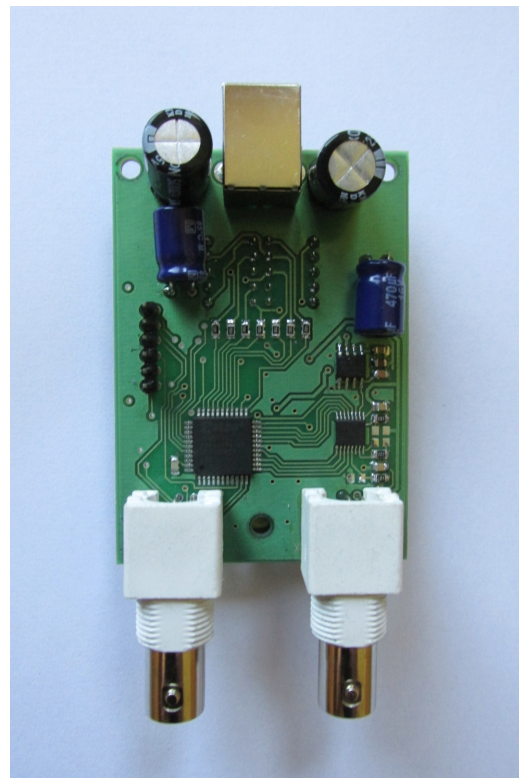
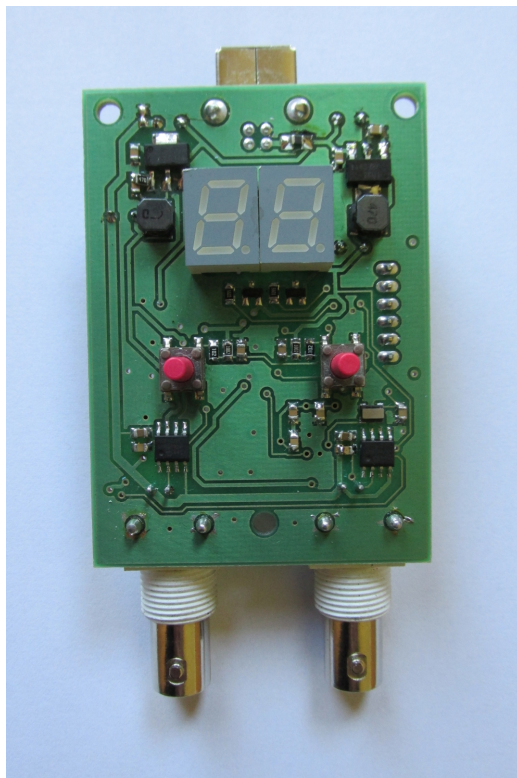
## C Rozpiska materiálu

Počet	Hodnota	Pouzdro	Součástka
1	DPS	XDORNA03_V7_X	DPS
1	0R	0805	R8
2	1k	0805	R9, R12
1	2k2	0805	R18
2	5k1	0805	R10, R13
4	10k	0805	R11, R15, R16, R30
7	1k5	0805	R1, R2, R3, R4, R5, R6, R7
1	15p	0805	C24
1	330p	0805	C22
15	100n	0805	C1, C2, C3, C4, C6, C8, C9, C10, C11, C16, C17, C18, C19, C20, C23
3	1u	0805	C5, C7, C21
2	470u/6V3	R3,5-8	C14, C15
2	1000u/10V	E3,5-8	C12, C13
1	BLM21PG	0805	FB1
2	47u	SRN5040	L1, L2
1	1x6	2,54	J1
1	XC9572XL	TQFP44	U1
1	MAX5102BEUE	TSSOP16	U2
1	MCP1501-12E	SOIC8	U3
2	LD1117S33	SOT223	IC1, IC2
2	MCP6021	SO8	IC3, IC4
1	ECS-327TXO	ECS-327TXO	XO1
2	BC807	SOT23	T1, T2
2	DTSM-65R-V-B	SWT6-M	SW1, SW2
2	SA39-12SRWA	HDSP-E	DISP1, DISP2
2	BNC	AMP_227161	X1, X2
1	USB	USB-B-PTH	USB
1	47R	0805	R31
1	celo_usb		Krabička 1/5
1	celo_bnc		Krabička 2/5
1	vrch		Krabička 3/5
1	spodek		Krabička 4/5
1	štítek		Krabička 5/5
6	M3x15		Šroub
3	M3x12		Distanční sloupek

## D Části modelu krabičky



## E Foto, osazená DPS a oživení



## F Zdrojové soubory VHDL

Výpis F.1: reseni.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reseni is
    Port (
        clock_from_TCX0 : in  STD_LOGIC;
        out_rectangular : out  STD_LOGIC;
        out_triangle : out  STD_LOGIC_vector (7 downto 0);
        out_7seg : out  STD_LOGIC_VECTOR (6 downto 0);
        out_pos : out  STD_LOGIC_VECTOR (1 downto 0);
        btn_up : in  STD_LOGIC;
        btn_down : in  STD_LOGIC--;
    );
end reseni;

architecture structural of reseni is

    signal sig_reset : STD_LOGIC;
    signal set_freq : STD_LOGIC_VECTOR (2 downto 0) := "000";
    signal sig_clk_core : STD_LOGIC;
    signal sig_signal_from_divider: STD_LOGIC;
    signal sig_btn_up, sig_btn_down : STD_LOGIC;

    component main_divider
        Port (
            main_divider_clk_in : in  STD_LOGIC;
            main_divider_clk_out_signal : out  STD_LOGIC;
            main_divider_clk_out_core : out  STD_LOGIC;
            main_divider_value : STD_LOGIC_VECTOR (2 downto 0)
        );
    end component;
```

<u>component</u> divider	35
<u>Port</u> (	36
divider_clk_in : <u>in</u> STD_LOGIC;	37
divider_clk_out : <u>out</u> STD_LOGIC;	38
divider_rst : <u>in</u> STD_LOGIC	39
);	40
<u>end component</u> ;	41
	42
<u>component</u> counter	43
<u>Port</u> (	44
counter_clk_in : <u>in</u> STD_LOGIC;	45
counter_out_tri : <u>out</u> STD_LOGIC_VECTOR (7 <u>downto</u> 0);	46
counter_rst : <u>in</u> STD_LOGIC	47
);	48
<u>end component</u> ;	49
	50
<u>component</u> seven_seg_decode	51
<u>Port</u> (	52
decoder_in : <u>in</u> STD_LOGIC_VECTOR (2 <u>downto</u> 0);	53
decoder_seg_out : <u>out</u> STD_LOGIC_VECTOR (6 <u>downto</u> 0);	54
decoder_pos : <u>out</u> STD_LOGIC_VECTOR (1 <u>downto</u> 0);	55
decoder_clk : <u>in</u> STD_LOGIC	56
);	57
<u>end component</u> ;	58
	59
<u>component</u> core	60
<u>Port</u> (	61
core_clk : <u>in</u> STD_LOGIC;	62
core_freq_out : <u>out</u> STD_LOGIC_VECTOR (2 <u>downto</u> 0);	63
core_rst_out : <u>out</u> STD_LOGIC;	64
core_btn_up : <u>in</u> STD_LOGIC;	65
core_btn_dwn : <u>in</u> STD_LOGIC	66
);	67
<u>end component</u> ;	68
	69
<u>component</u> debouncer	70
<u>Port</u> (	71
debouncer_in : <u>in</u> STD_LOGIC;	72
debouncer_out : <u>out</u> STD_LOGIC;	73
debouncer_clk : <u>in</u> STD_LOGIC	74
);	75
<u>end component</u> ;	76

<u>begin</u>	77
	78
	79
ent_main_divider : main_divider	80
<u>port map</u>	81
(clock_from_TCX0, sig_signal_from_divider, sig_clk_core,	82
set_freq);	83
ent_counter : counter	84
<u>port map</u>	85
(sig_signal_from_divider, out_triangle, sig_reset);	86
	87
ent_sig_divider : divider	88
<u>port map</u>	89
(sig_signal_from_divider, out_rectangular, sig_reset);	90
	91
ent_segment : seven_seg_decode	92
<u>port map</u>	93
(set_freq, out_7seg, out_pos, sig_clk_core);	94
	95
ent_core : core	96
<u>port map</u>	97
(sig_clk_core, set_freq, sig_reset, sig_btn_up, sig_btn_down);	98
	99
ent_deb_up : debouncer	100
<u>port map</u>	101
(btn_up, sig_btn_up, sig_clk_core);	102
	103
ent_deb_down : debouncer	104
<u>port map</u>	105
(btn_down, sig_btn_down, sig_clk_core);	106
	107
<u>end</u> Structural;	108



Výpis F.2: main\_divider.vhd

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_1164.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity main_divider is
7      Port (
8          main_divider_clk_in : in  STD_LOGIC;
9          main_divider_clk_out_signal : out  STD_LOGIC;
10         main_divider_clk_out_core : out  STD_LOGIC;
11         main_divider_value : STD_LOGIC_VECTOR (2 downto 0)
12     );
13 end main_divider;
14
15 architecture Behavioral of main_divider is
16
17     signal output: STD_LOGIC_VECTOR(7 downto 0) := (others => '0');
18
19 begin
20
21     process(main_divider_clk_in)
22     begin
23         if(rising_edge(main_divider_clk_in)) then
24             output <= output+'1';
25         end if;
26     end process;
27
28     main_divider_clk_out_core <= output(7);
29
30     with main_divider_value select
31     main_divider_clk_out_signal <= output(5) when "000", --1
32                                     output(4) when "001", --2
33                                     output(3) when "010", --4
34                                     output(2) when "011", --8
35                                     output(1) when "100", --16
36                                     output(0) when others; --32
37
38 end Behavioral;

```

<u>library</u> IEEE;	1
<u>use</u> IEEE.STD_LOGIC_1164. <u>ALL</u> ;	2
<u>use</u> IEEE.STD_LOGIC_1164. <u>ALL</u> ;	3
<u>use</u> IEEE.STD_LOGIC_ARITH. <u>ALL</u> ;	4
<u>use</u> IEEE.STD_LOGIC_UNSIGNED. <u>ALL</u> ;	5
	6
<u>entity</u> divider <u>is</u>	7
<u>Port</u> (	8
divider_clk_in : <u>in</u> STD_LOGIC;	9
divider_clk_out : <u>out</u> STD_LOGIC;	10
divider_rst : <u>in</u> STD_LOGIC	11
);	12
<u>end</u> divider;	13
	14
<u>architecture</u> Behavioral <u>of</u> divider <u>is</u>	15
	16
<u>signal</u> output: STD_LOGIC_VECTOR(8 <u>downto</u> 0) := "01111111";	17
	18
<u>begin</u>	19
	20
<u>process</u> (divider_clk_in,divider_rst)	21
<u>begin</u>	22
<u>if</u> (divider_rst='1') <u>then</u>	23
output<="01111111";	24
<u>elsif</u> (divider_clk_in'event <u>and</u> divider_clk_in='1') <u>then</u>	25
output <=output+'1';	26
<u>end if</u> ;	27
<u>end process</u> ;	28
	29
divider_clk_out <=   output(8);	30
	31
<u>end</u> Behavioral;	32

Výpis F.4: debouncer.vhd

<u>library</u> IEEE;	1
<u>use</u> IEEE.STD_LOGIC_1164. <u>ALL</u> ;	2
	3
<u>entity</u> debouncer <u>is</u>	4
<u>Port</u> (	5
debouncer_in : <u>in</u> STD_LOGIC;	6
debouncer_out : <u>out</u> STD_LOGIC;	7
debouncer_clk : <u>in</u> STD_LOGIC	8
);	9
<u>end</u> debouncer;	10
	11
<u>architecture</u> Behavioral <u>of</u> debouncer <u>is</u>	12
	13
<u>Signal</u> mem1, mem2, mem3: STD_LOGIC:= '1';	14
	15
<u>begin</u>	16
	17
<u>Process</u> (debouncer_clk)	18
<u>begin</u>	19
<u>if</u> (rising_edge(debouncer_clk)) <u>then</u>	20
mem3 <= mem2;	21
mem2 <= mem1;	22
mem1 <= debouncer_in;	23
<u>end if</u> ;	24
<u>end process</u> ;	25
	26
debouncer_out <= (mem1 <u>nor</u> mem2) <u>nand</u> mem3;	27
	28
<u>end</u> Behavioral;	29

Výpis F.5: counter.vhd

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity counter is
    Port (
        counter_clk_in : in  STD_LOGIC;
        counter_out_tri : out STD_LOGIC_VECTOR (7 downto 0);
        counter_rst : in  STD_LOGIC
    );
end counter;

architecture Behavioral of counter is

    signal direction : STD_LOGIC := '1';
    signal tmp : STD_LOGIC_VECTOR (7 downto 0) := "00000000";

begin

    process(counter_clk_in, counter_rst)
    begin
        if(counter_rst = '1') then
            tmp <= "00000000";
            direction <= '1';
        elsif(rising_edge(counter_clk_in)) then
            if(direction = '1' and tmp/="11111111") then
                tmp <= tmp+1;
            elsif(direction = '0' and tmp/="00000000") then
                tmp <= tmp-1;
            elsif(direction = '0' and tmp="00000000") then
                direction <= '1';
            elsif(direction = '1' and tmp="11111111") then
                direction<= '0';
            end if;
        end if;
    end process;

    counter_out_tri<=tmp;

end Behavioral;

```

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;

entity core is
  Port (
    core_clk : in STD_LOGIC;
    core_freq_out : out STD_LOGIC_VECTOR (2 downto 0);
    core_rst_out : out STD_LOGIC;
    core_btn_up : in STD_LOGIC;
    core_btn_dwn : in STD_LOGIC
  );
end core;

architecture Behavioral of core is
  signal sig_rst : STD_LOGIC := '1';
  signal tmp_mem : STD_LOGIC_VECTOR (2 downto 0) := "000";
  signal sig_up_flag1 : STD_LOGIC := '0';
  signal sig_down_flag1 : STD_LOGIC := '0';
  signal sig_up_flag2 : STD_LOGIC := '0';
  signal sig_down_flag2 : STD_LOGIC := '0';

begin

  process(core_btn_up, core_btn_dwn, core_clk)
  begin
    if(rising_edge(core_btn_dwn)) then
      sig_down_flag1 <= not sig_down_flag1;
    end if;
    if(rising_edge(core_btn_up)) then
      sig_up_flag1 <= not sig_up_flag1;
    end if;
  end process;

```

<u>if</u> (rising_edge(core_clk)) <u>then</u>	37
<u>if</u> (sig_rst ='1') <u>then</u>	38
sig_rst<='0';	39
<u>end if</u> ;	40
<u>if</u> (sig_down_flag1 /= sig_down_flag2) <u>then</u>	41
sig_down_flag2<=sig_down_flag1;	42
<u>if</u> (tmp_mem > "000") <u>then</u>	43
sig_rst <= '1';	44
tmp_mem<=tmp_mem-1;	45
<u>end if</u> ;	46
<u>end if</u> ;	47
<u>if</u> (sig_up_flag1 /= sig_up_flag2) <u>then</u>	48
sig_up_flag2<=sig_up_flag1;	49
<u>if</u> (tmp_mem < "101") <u>then</u>	50
sig_rst <= '1';	51
tmp_mem<=tmp_mem+1;	52
<u>end if</u> ;	53
<u>end if</u> ;	54
<u>end if</u> ;	55
<u>end process</u> ;	56
	57
core_rst_out <= sig_rst;	58
core_freq_out<=tmp_mem;	59
	60
<u>end Behavioral</u> ;	61

Výpis F.7: seven\_seg\_decode.vhd

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity seven_seg_decode is
    Port (
        decoder_in : in  STD_LOGIC_VECTOR (2 downto 0);
        decoder_seg_out : out STD_LOGIC_VECTOR (6 downto 0);
        decoder_pos : out STD_LOGIC_VECTOR (1 downto 0);
        decoder_clk : in STD_LOGIC
    );
end seven_seg_decode;

architecture Behavioral of seven_seg_decode is

    signal sig_decoder_pos : STD_LOGIC_VECTOR (1 downto 0):="01";

begin

    process(decoder_clk)
    begin
        if(rising_edge(decoder_clk)) then
            if (sig_decoder_pos="10" or sig_decoder_pos="01" ) then
                sig_decoder_pos<= not sig_decoder_pos;
            end if;
        end if;
    end process;

    with decoder_in & sig_decoder_pos select
    decoder_seg_out <= "1111111" when "000" & "10", --1 r display
                        "1111111" when "001" & "10" , --2
                        "1111111" when "010" & "10" , --4
                        "1111111" when "011" & "10" , --8
                        "0000010" when "100" & "10" , --16
                        "0100100" when "101" & "10" , --32
                        "1111001" when "000" & "01" , --1 l display
                        "0100100" when "001" & "01" , --2
                        "0011001" when "010" & "01" , --4
                        "0000000" when "011" & "01" , --8
                        "1111001" when "100" & "01" , --16
                        "0110000" when "101" & "01" , --32
                        "1111111" when others;
    decoder_pos<=sig_decoder_pos;

    end Behavioral;

```

## G Obsah CD

/	..... kořenový adresář přiloženého CD
BP.pdf	..... text práce
DPS	..... vše okolo dps
xdorna03_V_7_V.brd	
xdorna03_V_7_V.sch	
BOM.xls	
dps_top.PNG	
dps_bottom.PNG	
osazeni_top.PNG	
osazeni_bottom.PNG	
Krabicka	..... model krabičky
celo_bnc.SLDPRT	
celo_usb.SLDPRT	
spodek.SLDPRT	
vrch.SLDPRT	
stitek_grafika.eps	
VHDL	..... zdrojové soubory
core.VHD	
counter.VHD	
debouncer.VHD	
divider.VHD	
main_divider.VHD	
reseni.VHD	
reseni.jed	
seven_seg_decode.VHD	